Meta-Learning Hypothesis Spaces for Sequential Decision-making

Parnian Kassraie [1] **Jonas Rothfuss** [1] **Andreas Krause** [1]

Abstract

Obtaining reliable, adaptive confidence sets for prediction functions (hypotheses) is a central challenge in sequential decision-making tasks, such as bandits and model-based reinforcement learning. These confidence sets typically rely on prior assumptions on the hypothesis space, e.g., the *known* kernel of a Reproducing Kernel Hilbert Space (RKHS). Hand-designing such kernels is error prone, and misspecification may lead to poor or unsafe performance. In this work, we propose to *meta-learn a kernel from offline data* (META-KEL). For the case where the unknown kernel is a combination of known base kernels, we develop an estimator based on structured sparsity. Under mild conditions, we guarantee that our estimated RKHS yields valid confidence sets that, with increasing amounts of offline data, become *as tight as those given the true unknown kernel*. We demonstrate our approach on the kernelized bandit problem (a.k.a. Bayesian optimization), where we establish regret bounds competitive with those given the true kernel. We also empirically evaluate the effectiveness of our approach on a Bayesian optimization task.

1. Introduction

A number of well-studied machine learning problems such as bandits, Bayesian optimization (BO) and model-based reinforcement learning are characterized by an agent that sequentially interacts with an unknown, responsive system. Throughout the interaction, the agent's goal is to maximize the cumulative reward based on an unknown underlying function f. Common to such sequential decision-making problems is an exploration-exploitation trade-off. That is, the agent needs to optimize its reward while, at the same time, learns more about the unknown function f. Confidence sets capture and quantify the uncertainty of the learner about f. Thus, they are an integral tool for directing exploration towards areas of high uncertainty and balancing it against exploitation. Moreover, in safety-critical applications, confidence sets are used to reason about the safety of actions. Thus, they are central to efficiency and safety of exploration. In theoretical analysis of sequential decision-making algorithms, a common assumption is that f resides in an RKHS with a *known* kernel function. This assumption allows for the construction of the confidence sets.

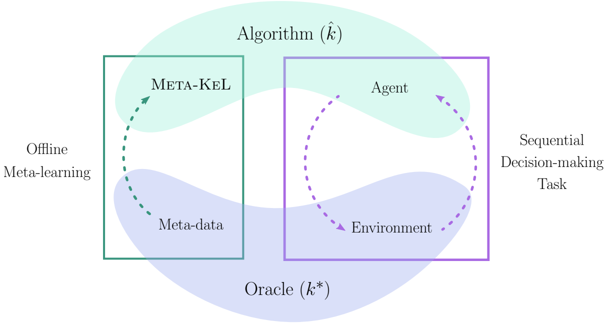


Figure 1. Overview of the described framework with k^* as the true kernel function and $\hat{k}$ as the solution to META-KEL.

In practice, however, the true kernel is unknown and needs to be hand-crafted based on the problem instance. This is a delicate task, since the hand-crafted hypothesis space has to contain the unknown target function f. If this is not the case, the learner may be over-confident and converge to a sub-optimal policy, or incorrectly classify actions as safe. At the same time, we want the chosen hypothesis space to be as small so that the variance of the associated learner is low and the agent converges quickly. This constitutes a dilemma, where we need to trade off efficiency with a potential loss in consistency.

We approach this dilemma in a data-driven manner. Many applications of sequential decision-making, such as hyper-parameter tuning with BO or online nonlinear control, are of repetitive nature. Often, there is available data from similar but not identical tasks which have been solved before. Therefore, we propose to *meta-learn the kernel function, and thus the RKHS, from offline meta-data*. Our method, *Meta-Kernel Learning* (META-KEL), works with a generic (i.e., not necessarily i.i.d.) data model and may be applied to a variety of sequential decision-making tasks.

[1] ETH Zurich, Switzerland. Correspondence to: Parnian Kassraie <pkassraie@ethz.ch>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

We formally analyze the problem when the true kernel is a combination of known base kernels. We prove that the solution to META-KEL corresponds to an RKHS which contains the true function space (Theorem 4.3). Further, the meta-learned kernel has a sparse structure (Proposition 4.4) which reduces the variance of the resulting learner, and makes the learner more efficient for solving the downstream sequential decision-making problem. With mild assumptions on the data, we show that the meta-learned kernel yields anytime valid confidence bands matching the ones given oracle knowledge of the true kernel, as more samples of the meta-data are provided (Theorem 5.1). These provably reliable confidence estimates constitute the key contribution of our work and distinguishes META-KEL from prior attempts.

To demonstrate how META-KEL can be applied to a sequential task, we analyze a Bayesian optimization algorithm when it uses the meta-learned kernel and compare it to the same algorithm when it has knowledge of the true kernel, i.e., the oracle algorithm. By increasing size of the meta-learning data, the regret bound we obtain approaches the rate of the oracle (Corollary 5.2).

Contributions Our main contributions are:

- We introduce META-KEL, a method for meta-learning the hypothesis space of a sequential decision task, which yields provably valid adaptive confidence sets.
- Our meta-learnt confidence bounds converge to the ones estimated by the oracle at a $\mathcal{O}(1/\sqrt{mn})$ rate. Here, m is the number of tasks in the meta-data and n is the number of samples per task.
- Applied to BO, our results imply a sublinear regret guarantee for the GP-UCB algorithm using our meta-learned kernel. This bound approaches that of the oracle algorithm as the amount of meta-data increases.

2. Related Work

Numerous sequential decision-making methods rely on confidence sets for uncertainty quantification, e.g., UCB algorithms (Srinivas et al., 2010; Chowdhury & Gopalan, 2017) for Bayesian optimization and bandits, safe exploration and various forms of RL (Berkenkamp et al., 2017; Curi et al., 2020; Kakade et al., 2020; Sessa et al., 2020). Most of these methods assume the true hypothesis space as given. However, in practice, we typically do not know the correct hypothesis space, e.g., in form of a kernel. A body of recent work considers the unknown kernel setting and analyzes the effect of working with a misspecified hypothesis space (Wynne et al., 2021; Simchowitz et al., 2021; Bogunovic & Krause, 2021). Alternatively, Wang & de Freitas (2014) and Berkenkamp et al. (2019) propose to successively expand the hypothesis space throughout the course of BO so that the algorithm remains no-regret in a setting where the kernel lengthscale is unknown.

Our work relates to meta-learning for Bayesian optimization. There is a recent line of algorithms that improve accuracy of base sequential learners via meta-learning, albeit without theoretical guarantees (Rothfuss et al., 2021a;b), or with mild guarantees in special cases (Kveton et al., 2020; Boutilier et al., 2020). There are a number of results on updating bandit priors or policies by meta-learning, under problem settings different than ours. Basu et al. (2021) work with a sequence of multi-armed bandit tasks, and adaptively meta-learn the *mean* of the Gaussian prior used for the next task. Others consider solving a number of structurally similar linear bandit tasks in parallel (Wang et al., 2017; Cella et al., 2020; Cella & Pontil, 2021). They propose how to efficiently update the policy when each learner has access to the data across all tasks. We give stronger guarantees in less restrictive setting, compared to Wang et al. (2018b) which analyzes the simple regret of the GP-UCB algorithm (Srinivas et al., 2010) for multi-armed and linear bandits, when the mean and variance of the Gaussian prior are unknown, and there is sufficient offline i.i.d. data drawn from the same Gaussian distribution.

Our framework considers structural sparsity at the kernel level, translating to group sparsity for the coefficients vectors, if applied to linear bandits. Thus our work relates to results on sparse linear bandits and Lasso bandits. In this area, Bastani & Bayati (2020) and Hao et al. (2020) give dimension-independent regret bounds for *Explore-Then-Commit* algorithms under certain assumptions over the action set. This work does not consider offline data.

We draw inspiration from the early Multiple Kernel Learning (MKL) literature, which focuses on kernel design for classification with SVMs (Bach et al., 2004; Gönen & Alpaydın, 2011; Kloft et al., 2011; Evgeniou & Pontil, 2004; Cristianini et al., 2006; Ong et al., 2005). In contrast, our key contribution is to derive adaptive confidence bounds from meta-learnt kernels for regression, even for non-i.i.d. data. Orthogonal to most prior works, we reduce the kernel learning problem to group Lasso and leverage the properties of the Lasso estimator, in particular seminal results of Lounici et al. (2011) and Bach (2008). Other relevant works on convergence properties of the group Lasso include Koltchinskii & Yuan (2008), Liu & Zhang (2009) and Bunea et al. (2013).

3. Problem Statement

Consider a sequential decision-making problem, where the agent repeatedly interacts with an environment and makes observations

$$\boldsymbol{y}_t = f(\boldsymbol{x}_t) + \varepsilon_t \tag{1}$$

of an unknown function $f : \mathcal{X} \to \mathbb{R}$ residing in an RKHS $\mathcal{H}_{k^*}$ that corresponds to an *unknown* kernel function k^*.[1] We further assume that the function has a bounded kernel norm $\|f\|_{k^*} \leq B$ and that the domain $\mathcal{X} \subset \mathbb{R}^{d_0}$ is compact. The observation noise ε_t are i.i.d. samples from a zero-mean sub-Gaussian distribution with variance proxy σ^2. At every step t, the chosen input $\boldsymbol{x}_t$ only depends on the history up to step t, denoted by the random sequence $H_{t-1} = \{(\boldsymbol{x}_\tau, y_\tau) : 1 \leq \tau \leq t-1\}$. No further assumptions are made about the algorithm or the policy for choosing $\boldsymbol{x}_t$. Depending on the application, Equation (1) can serve different purposes: It can describe the stochastic reward model of a bandit problem, or it may be the transition dynamics of an RL agent in a stochastic environment.

For solving such problems, a central prerequisite for numerous algorithms are *confidence sets* for $f(\boldsymbol{x})$ based on the history H_{t-1} to balance exploration and exploitation at any step t. For any $\boldsymbol{x} \in \mathcal{X}$, the set $\mathcal{C}_{t-1}(\boldsymbol{x})$ defines an interval to which $f(\boldsymbol{x})$ belongs with high probability such that,

$$\mathbb{P}\left(\forall \boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) \in \mathcal{C}_{t-1}(\boldsymbol{x})\right) \geq 1 - \delta.$$

The center of this interval reflects the current knowledge of the agent, relevant for exploitation, and the width corresponds to the uncertainty, guiding further exploration. When the true kernel is known, an approach commonly used in the kernelized bandit literature (Abbasi-Yadkori et al., 2011; Srinivas et al., 2010; Russo & Van Roy, 2014) is to build sets of the form

$$\begin{aligned}\mathcal{C}_{t-1}(k; \boldsymbol{x}) = [&\mu_{t-1}(k; \boldsymbol{x}) - \nu_t \sigma_{t-1}(k; \boldsymbol{x}), \\ &\mu_{t-1}(k; \boldsymbol{x}) + \nu_t \sigma_{t-1}(k; \boldsymbol{x})]\end{aligned} \tag{2}$$

where the exploration coefficient ν_t depends on the desired confidence level $1-\delta$, and may be set based on the objective of the decision-making task. The functions μ_{t-1} and σ_{t-1} set the center and width of the confidence set as

$$\begin{aligned}\mu_{t-1}(k; \boldsymbol{x}) &= \boldsymbol{k}_{t-1}^T(\boldsymbol{x})(\boldsymbol{K}_{t-1} + \bar{\sigma}^2 \boldsymbol{I})^{-1} \boldsymbol{y}_{t-1} \qquad (3)\\ \sigma_{t-1}^2(k; \boldsymbol{x}) &= k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_{t-1}^T(\boldsymbol{x})(\boldsymbol{K}_{t-1} + \bar{\sigma}^2 \boldsymbol{I})^{-1} \boldsymbol{k}_{t-1}(\boldsymbol{x})\end{aligned}$$

where $\bar{\sigma}$ is a constant, $\boldsymbol{y}_{t-1} = [y_\tau]_{\tau<t}$ is the vector of observed values, $\boldsymbol{k}_{t-1}(\boldsymbol{x}) = [k(\boldsymbol{x}, \boldsymbol{x}_\tau)]_{\tau<t}$, and $\boldsymbol{K}_{t-1} = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j<t}$ is the kernel matrix. Hence working with the right kernel function plays an integral role in constructing well-specified sets. Since, in practice, the true kernel k^* is not known by the learner, most approaches use a hand-designed kernel that suits the problem instance at hand or conservatively pick an expressive kernel that constructs a rich RKHS which is very likely to contain f. There are a number of empirical approaches for selecting the kernel, e.g., cross-validation or maximizing the marginal likelihood. However, such methods tend to be unreliable when the available data is non-i.i.d. and comes from sequential learning tasks.

Addressing the issue of selecting a correct and yet efficient kernel, we pursue a data-driven approach and *meta-learn a kernel that provably yields valid confidence intervals*. This guarantee is valid regardless of how the meta-data is gathered, as long as it satisfies some basic conditions discussed later in Assumptions 3.1 and 4.2. We consider an offline collection of datasets $\mathcal{D}_{n,m} = \{(\boldsymbol{x}_{s,i}, y_{s,i})_{i \leq n}\}_{s \leq m}$ from m possibly non-i.i.d. tasks, each with a sample size n. Suppose, for each task s, labels are generated by

$$y_{s,i} = f_s(\boldsymbol{x}_{s,i}) + \varepsilon_{s,i} \tag{4}$$

for $i \leq n$, where $\varepsilon_{s,i}$ are zero-mean i.i.d. sub-Gaussian noise with variance proxy σ^2. We assume the tasks are related by the fact that all $f_s : \mathcal{X} \to \mathbb{R}$ come from the same function class $\mathcal{H}_{k^*}$ and have a bounded RKHS norm $\|f_s\|_{k^*} \leq B$. We do not make any assumptions on the policy based on which the points $\boldsymbol{x}_{s,i}$ are chosen.

Assumptions Our analysis requires some assumptions on the kernel function. In particular, we assume that k^* is a finite combination of known base kernels,

$$k^*(\boldsymbol{x}, \boldsymbol{x}') = \sum_{j=1}^{p} \eta_j^* k_j(\boldsymbol{x}, \boldsymbol{x}'), \tag{5}$$

where the weight vector $\boldsymbol{\eta}^* \geq 0$ is *unknown*. Without loss of generality, we assume that k^* and the base kernels are all *normalized*, i.e., $\|\boldsymbol{\eta}^*\|_1 \leq 1$ and $k_j(\boldsymbol{x}, \boldsymbol{x}') \leq 1$ for all $1 \leq j \leq p$ and $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$. The weight vector $\boldsymbol{\eta}^*$ is potentially *sparse*, since not all the candidate kernels k_j actively contribute to the construction of k^*. We use $J_{k^*} = \{1 \leq j \leq p : \eta_j^* \neq 0\}$ to refer to the group of base kernels that are present in k^*. The sparse construction of k^* imposes favorable structure on the meta-data, which essentially allows us to meta-model-select the hypothesis space and recover the true sparsity pattern denoted by J_{k^*}. We further assume that each k_j has a d_j-dimensional feature map, i.e., $k_j(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}_j^T(\boldsymbol{x})\boldsymbol{\phi}_j(\boldsymbol{x}')$, where $\boldsymbol{\phi}_j \in \mathbb{R}^{d_j}$. For the scope of this paper, we assume that $d_{\max} < \infty$, where $d_{\max} := \max_{j \leq p} d_j$. In this finite regime, the analysis can also be carried out in a finite-dimensional vector space. Nevertheless, we use a function space notation since, even though our theory focuses on the finite-dimensional setting, empirically our approach is also applicable to kernels with infinite dimensional feature map.[2]

Let $\boldsymbol{\phi}(\boldsymbol{x})$ denote the d-dimensional feature map for k^* where $d = \sum_{j=1}^{p} d_j$ and

[1] Appendix A.1 presents a compact refresher on the RKHS.

[2] Meta-learning the hypothesis space in the $p \to \infty$ limit will be challenging. However, we expect to obtain an extension to infinite dimensional base-kernels in future work.

$$\boldsymbol{\phi}(\boldsymbol{x}) = \left(\sqrt{\eta_1^*}\boldsymbol{\phi}_1^T(\boldsymbol{x}), \cdots, \sqrt{\eta_p^*}\boldsymbol{\phi}_p^T(\boldsymbol{x})\right)^T.$$

For each task s, the function f_s is contained in $\mathcal{H}_{k^*}$. By Mercer's theorem f_s may be decomposed as

$$f_s(\boldsymbol{x}) = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{\beta}_s^* = \sum_{j=1}^{p}\sqrt{\eta_j^*}\boldsymbol{\phi}_j^T(\boldsymbol{x})\boldsymbol{\beta}_s^{*(j)}, \qquad (6)$$

where $\boldsymbol{\beta}_s^* \in \mathbb{R}^d$ is the coefficients vector of task s and $\boldsymbol{\beta}_s^{*(j)} \in \mathbb{R}^{d_j}$ is the sub-vector corresponding to kernel k_j. It is not possible to meta-select a base kernel k_j which has not contributed to the generation of the meta-data. Therefore, if a base kernel is active in the construction of $\mathcal{H}_{k^*}$, it is only natural to assume that there is some task in the meta-data which reflects this presence. More formally, we assume that, for any $j \in J_{k^*}$, there exists some $s \leq m$ where $\boldsymbol{\beta}_s^{*(j)} \neq 0$. We define $\boldsymbol{\beta}^* = (\boldsymbol{\beta}_1^{*T}, \cdots, \boldsymbol{\beta}_m^{*\,T})^T \in \mathbb{R}^{md}$ as the concatenated coefficients vector for all tasks. To refer to the group of coefficients that correspond to kernel k_j across all tasks, we use $\boldsymbol{\beta}^{*(j)} = ((\boldsymbol{\beta}_1^{*(j)})^T, \cdots, (\boldsymbol{\beta}_m^{*(j)})^T)^T \in \mathbb{R}^{md_j}$. Table 1 presents a compact guide to the notation introduced here. Our next assumption guarantees that the meta-learning problem is not ill-posed.

Assumption 3.1 (Group Beta-min Condition)**.** There exists $c_1 > 0$ s.t. for all $j \in J_{k^*}$ it holds that $\left\|\boldsymbol{\beta}^{*(j)}\right\|_2 \geq c_1$.

This assumption is inevitable for recovering the sparsity pattern from empirical data and it is widely used in the high-dimensional statistics literature (e.g., Bühlmann & Van De Geer, 2011; Zhao & Yu, 2006; Van de Geer et al., 2011). Assumption 3.1 implies that for j to be in J_{k^*}, the coefficients vector corresponding to kernel k_j can not be zero or arbitrarily close to zero. In practice, $\left\|\boldsymbol{\beta}^{*(j)}\right\|_2$ has to be comparable with the noise level for the activity of a base kernel not to be mistaken with randomness.

4. Meta-learning the Hypothesis Space (META-KEL)

In the following section, we present our formulation of the meta-learning problem and analyze the properties of the learned hypothesis space. We meta-learn the kernel by solving the following optimization problem. Then, we set the hypothesis space of the downstream learning algorithm to be the RKHS of the meta-learned kernel.

$$\min_{\boldsymbol{\eta}, f_1, \ldots, f_m} \frac{1}{m}\sum_{s=1}^{m}\left[\frac{1}{n}\sum_{i=1}^{n}\left(y_{s,i} - f_s(\boldsymbol{x}_{s,i})\right)^2\right] + \frac{\lambda}{2}\sum_{s=1}^{m}\|f_s\|_k^2 + \frac{\lambda}{2}\|\boldsymbol{\eta}\|_1 \qquad (7)$$

$$\text{s.t. } \forall s: f_s \in \mathcal{H}_k,\ k = \sum_{j=1}^{p}\eta_j k_j, 0 \leq \boldsymbol{\eta}$$

We will refer to this problem as *Meta-Kernel Learning* (META-KEL). The first part of the objective is similar to the kernel ridge regression loss, and accounts for how well a series of regularized f_s fit the meta-data. The last term regularizes our choice of the kernel function. We use ℓ_1-norm regularization for $\boldsymbol{\eta}$ to implicitly perform meta-model-selection. As shown in Proposition 4.4, the meta-learned kernel will reflect the sparsity pattern of the true kernel. The optimization problem (7) is convex and admits an efficient solution, as explained next.

We first introduce a vectorized formulation of Equation (7). Let $\boldsymbol{y}_s \in \mathbb{R}^n$ denote the observed values for a task s and $\boldsymbol{y} = (\boldsymbol{y}_1^T, \cdots, \boldsymbol{y}_m^T)^T \in \mathbb{R}^{mn}$ the multi-task stacked vector of observations. We then design a multi-task feature matrix. We define $\boldsymbol{\Phi}$ to be a $mn \times md$ block-diagonal matrix, where each block s corresponds to $\boldsymbol{\Phi}_s = (\boldsymbol{\phi}(\boldsymbol{x}_{s,1}), \cdots, \boldsymbol{\phi}(\boldsymbol{x}_{s,n}))^T$, the $n \times d$ feature matrix of task s. Figure 7 provides an illustration thereof. As shown in Proposition 4.1, this vectorized design brings forth a parametric equivalent of META-KEL, which happens to be the well-known Group Lasso problem.

Proposition 4.1 (Solution of META-KEL)**.** *Let $k = \sum_j \hat{\eta}_j k_j$ be a solution to Problem (7). Then, for all $1 \leq j \leq p$, it holds that*

$$\hat{\eta}_j = \left\|\hat{\boldsymbol{\beta}}^{(j)}\right\|_2$$

with $\hat{\boldsymbol{\beta}} = (\hat{\boldsymbol{\beta}}^{(j)})_{j \leq p}$ as the solution of the following convex optimization problem:

$$\min_{\boldsymbol{\beta}} \frac{1}{mn}\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\beta}\|_2^2 + \lambda\sum_{j=1}^{p}\left\|\boldsymbol{\beta}^{(j)}\right\|_2. \qquad (8)$$

We show this equivalence by eliminating $\boldsymbol{\eta}$. We use a trick introduced by Bach et al. (2004), which, for $w, v \in \mathbb{R}$ states $2|w| = \min_{v \geq 0} w^2/v + v$. The proof is given in Appendix A.2. Problem (8) can be optimized by any Group Lasso solver. Bach et al. (2012) present a number of coordinate descent algorithms which efficiently find the solution.

Before introducing the meta-learned kernel $\hat{k}$, we note that Reproducing Kernel Hilbert Spaces are equivalent up to scaling of the kernel function. For $c > 0$, both $\mathcal{H}_k$ and the scaled version $\mathcal{H}_{ck}$ contain the same set of functions. Going from $\mathcal{H}_k$ to $\mathcal{H}_{ck}$, the RKHS norm of any member f would scale by $1/c$, i.e. $\|f\|_k = c\|f\|_{ck}$. Hence, the norm $\|\hat{\boldsymbol{\eta}}\|_1$ will be irrelevant when meta-learning the function space. This norm can be scaled or normalized, and still yield the same hypothesis space, only with a scaled operator norm. For consistency of notation, we define $\hat{k}$ as follows. For any two points $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$, set

$$\hat{k}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{j=1}^{p}\frac{\hat{\eta}_j}{c_1}\boldsymbol{\phi}_j^T(\boldsymbol{x})\boldsymbol{\phi}_j(\boldsymbol{x}'), \qquad (9)$$

where c_1 is the same constant as in Assumption 3.1. We emphasize that this scaling does not impose a new assumption on the problem as it is done only to simplify theorem statements. We denote the set of base kernels active in $\hat{k}$ with $J_{\hat{k}} = \{1 \leq j \leq p : \hat{\eta}_j \neq 0\}$. The meta-learned hypothesis space will then be $\mathcal{H}_{\hat{k}}$, the RKHS which corresponds to $\hat{k}$.

Properties of the Meta-learned Hypothesis Space The meta-learned $\hat{k}$ can be used as the kernel function for a model-based sequential decision making algorithm, as illustrated in Figure 1. Our goal is to analyze how this choice of kernel affects the success of the algorithm, compared to the oracle algorithm with access to the unknown kernel. To this end, we discuss properties of $\hat{k}$.

For our main result, we require a final technical assumption to ensure that our meta-data is sufficiently diverse. Consider some vector $\boldsymbol{b} \in \mathbb{R}^{md}$ that adheres to the same group structure as $\boldsymbol{\beta}^*$. For a set of group indices $J \subset \{1, \cdots, p\}$, we define $\boldsymbol{b}_J := (\boldsymbol{b}^{(j)})_{j \in J}$ as the sub-vector indicated by the groups in J.

Assumption 4.2 (Relaxed Sufficient Exploration)**.** There exists $\kappa = \kappa(s) > 0$ that satisfies

$$\kappa \leq \min_{J, \boldsymbol{b}} \frac{\|\Phi \boldsymbol{b}\|_2}{\sqrt{mn}\|\boldsymbol{b}_J\|_2},$$
$$\text{s.t.} \sum_{j \notin J} \left\|\boldsymbol{b}^{(j)}\right\|_2 \leq 3 \sum_{j \in J} \left\|\boldsymbol{b}^{(j)}\right\|_2, \ \boldsymbol{b} \neq 0, \ |J| \leq s.$$

This condition makes sure that the meta-training data is not degenerate, e.g., no two points $\boldsymbol{x}_{s,i}$ and $\boldsymbol{x}_{s,j}$ in the meta-data are identical or too close. Assumption 4.2 is immediately fulfilled if the minimum eigenvalue of $\boldsymbol{\Phi}$ is positive, i.e., if there exists some $\kappa > 0$ where $\lambda_{\min}(\boldsymbol{\Phi}) \geq \kappa$. This stronger version is sometimes referred to as Sufficient Exploration (Basu et al., 2021; Zhou et al., 2020). In the Lasso literature, Assumption 4.2 is commonly referred to as Restricted Eigenvalue Assumption (Bickel et al., 2009; Bühlmann & Van De Geer, 2011; Javanmard & Montanari, 2014). It is also often required to hold for the kernel matrix in the sparse linear bandits literature (Bastani & Bayati, 2020; Wang et al., 2018a; Hao et al., 2020; Kim & Paik, 2019). Lastly, Assumption 4.2 is satisfied with high probability if $\boldsymbol{\Phi}$ is an i.i.d. sub-Gaussian random matrix, since the eigenvalues of such random matrices are bounded away from zero (Vershynin, 2018, Theorem 4.6.1).

Theorem 4.3 (Hypothesis Space Recovery)**.** *Set* $0 \leq \delta \leq 1$, *and choose* λ *such that,*

$$\lambda \geq \frac{4\sigma}{\sqrt{mn}} \sqrt{1 + \frac{2}{m}\left(\log(2p/\delta) + \sqrt{md_{\max}\log(2p/\delta)}\right)}.$$

If $|J_{k^*}| \leq s$ *and Assumption 4.2 holds with* $\kappa(s)$*, then* $\mathcal{H}_{k^*} \subseteq \mathcal{H}_{\hat{k}}$ *and*

$$\|f\|_{\hat{k}} \leq \|f\|_{k^*}\Big(1 + \epsilon(n, m) + o\big(\epsilon\,(n, m)\,\big)\Big),$$

with probability greater than $1 - \delta$*, if* n *and* m *are large enough to satisfy* $\epsilon(n, m) \leq c_1$*. The absolute constant* c_1 *is defined in Assumption 3.1 and*

$$\epsilon(n, m) := \frac{32\sigma s}{\kappa^2(s)\sqrt{mn}} \times \sqrt{1 + \frac{2}{m}\left(\log(2p/\delta) + \sqrt{md_{\max}\log(2p/\delta)}\right)}.$$

The proof is given in Appendix B.1. Theorem 4.3 states that, provided enough meta-data, $\mathcal{H}_{k^*}$ is contained in $\mathcal{H}_{\hat{k}}$ with high probability and δ, the probability of failure in recovery, decreases as m and n grow (See Appendix B.3). In addition the RKHS norm of any $f \in \mathcal{H}_{k^*}$, can be bounded arbitrary well by $\|f\|_{\hat{k}}$, since the error $\epsilon(n, m)$ decreases at a $\mathcal{O}(s/\sqrt{mn}\sqrt{1 + m^{-1}\log p})$ rate. This matches the tightest rate for coordinate-wise convergence of the Group Lasso estimator under the same set of assumptions (Lounici et al., 2011; Bunea et al., 2013). The theorem implies that the meta-learner benefits more from increasing the number m of meta-data tasks, rather than increasing the sample size n of each task, since $\epsilon(n, m)$ shrinks faster with m compared to n. Note that increasing either m or n will result in convergence and therefore this theorem also holds for the classic offline kernel learning setup when the dataset consists of a single learning task ($m = 1$).

The Benefit of Structural Sparsity Consider a conservative hand-picked kernel function

$$k_{\text{full}} = 1/p \sum_{j=1}^{p} \boldsymbol{\phi}_j^T(\boldsymbol{x})\boldsymbol{\phi}_j(\boldsymbol{x}'), \tag{10}$$

which does not use any meta-data and instead incorporates all the considered base kernels. When p and $d_{\max}$ are finite, $\mathcal{H}_{k^*}$ is contained in $\mathcal{H}_{k_{\text{full}}}$ and the hand-picked hypothesis space is not misspecified. However, working with an overly large hypothesis space has downsides. Consider using k_{full} to estimate a function $f \in \mathcal{H}_{k^*}$. Then every base kernel, including k_j with $j \notin J_{k^*}$, appears in the construction of the estimator. These terms contribute to the estimation error and increase the variance of the function estimate. This slows down the rate of convergence, compared to the case where only active k_j are present in the kernel function. By meta-learning $\hat{k}$ via META-KEL, we can eliminate irrelevant candidate kernels and produce a structurally sparse hypothesis space. Proposition 4.4 guarantees this property. Its proof is given in Appendix B.2.

Proposition 4.4 (Bound on structural sparsity of $\hat{k}$)**.** *Set* $0 < \delta < 1$ *and choose* λ *according to Theorem 4.3. Let*

$|J_{k^*}| \leq s$ be the number candidate kernels that contribute to k^*. If Assumption 4.2 holds with $\kappa(s)$, then with probability greater than $1-\delta$, the number of kernels active in $\hat{k}$ is bounded by

$$|J_{\hat{k}}| \leq \frac{64s}{mn\kappa^2(s)}$$

which implies that if $mn > \frac{64s}{p\kappa^2(s)}$, then with the same probability

$$\mathcal{H}_{\hat{k}} \subsetneq \mathcal{H}_{k_{\text{full}}}.$$

Hence, in the presence of enough meta-data, $\mathcal{H}_{\hat{k}}$ is a strict subset of $\mathcal{H}_{k_{\text{full}}}$, and therefore

$$\mathcal{H}_{k^*} \overset{\text{w.h.p.}}{\subseteq} \mathcal{H}_{\hat{k}} \overset{\text{w.h.p.}}{\subsetneq} \mathcal{H}_{k_{\text{full}}}$$

where the left relation is due to Theorem 4.3. Figure 2 illustrates the nested sets. We conclude that our meta-learned hypothesis space has favorable properties: it contains the true hypothesis space, and it is sparse in structure, in particular, smaller than the conservative candidate space.

The fact that $\mathcal{H}_{\hat{k}}$ is smaller than $\mathcal{H}_{k_{\text{full}}}$ reduces the complexity of the downstream learning problem and yields faster convergence rates. We provide an example of this effect in Section 5, where we analyze a Bayesian optimization problem, and establish how choosing $\hat{k}$ improves upon k_{full}. Finally, our experiments (e.g. Figure 4) support the claim that in practice the BO algorithm is faster in finding the optimum when it uses the meta-learned kernel.

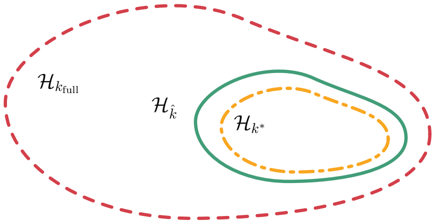


Figure 2. The oracle $\mathcal{H}_{k^*}$ (Eq. 5), the meta-learned $\mathcal{H}_{\hat{k}}$ (Eq. 9) and the hand-picked $\mathcal{H}_{k_{\text{full}}}$ (Eq. 10) hypothesis spaces (informal)

5. Sequential Decision-making with META-KEL

We now analyze the effect of using $\hat{k}$ as kernel function in the downstream sequential decision-making problem. We adopt the common construction of confidence sets given in Equation (2), and define $\hat{\mathcal{C}}_{t-1}(\boldsymbol{x}) := \mathcal{C}_{t-1}(\hat{k};\boldsymbol{x})$. We let $\hat{\mu}_{t-1}(\boldsymbol{x}) := \mu_{t-1}(\hat{k};\boldsymbol{x})$, and $\hat{\sigma}_{t-1}(\boldsymbol{x}) := \sigma_{t-1}(\hat{k};\boldsymbol{x})$, where $\mu_{t-1}(k;\boldsymbol{x})$ and $\sigma_{t-1}(k;\boldsymbol{x})$ are as defined in Equation (3), with time-varying $\bar{\sigma}^2 = 1+2/t$.[3]

Theorem 5.1 shows that for the right choice of ν_t, the set $\hat{C}_{t-1}(\boldsymbol{x})$ is a valid confidence bound for any $f \in \mathcal{H}_{k^*}$, evaluated at any $\boldsymbol{x} \in \mathcal{X}$, at any step t, with high probability.

Theorem 5.1 (Any-time Valid Confidence Bounds with META-KEL). *Let $f \in \mathcal{H}_{k^*}$ with $\|f\|_{k^*} \leq B$, where k^* is unknown. Under the assumptions of Theorem 4.3, with probability greater than $1-\delta$, for all $\boldsymbol{x} \in \mathcal{X}$ and all $t \geq 1$,*

$$|\hat{\mu}_{t-1}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \hat{\sigma}_{t-1}(\boldsymbol{x}) \Bigg(B\left(1+\frac{\epsilon(n,m)}{2c_1}\right) + \sigma\sqrt{\hat{d}\log\left(1+\frac{\bar{\sigma}^{-2}t}{c_1}\right)+2+2\log(1/\delta)}\Bigg)$$

where $\hat{d} = \sum_{j\in J_{\hat{k}}} d_j$ and $\bar{\sigma}^2 = 1+2/t$.

The proof is given in Appendix C. As discussed in Section 4, the $\epsilon(n,m)/2c_1$ term shrinks faster than $\mathcal{O}(1/\sqrt{mn})$ and $\hat{d}$ approaches $d^* = \sum_{j\in J_{k^*}} d_j$ at a similar rate. Therefore, Theorem 5.1 presents a tight confidence bound relative to the case when k^* is known by the agent. In this case, due to Chowdhury & Gopalan (2017), Theorem 2, the $1-\delta$ confidence bound would be,

$$|\mu_{t-1}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \sigma_{t-1}(\boldsymbol{x}) \Big(B + \sigma\sqrt{d^*\log\left(1+\bar{\sigma}^{-2}t\right)+2+2\log(1/\delta)}\Big).$$

where the mean and variance functions are defined by $\mu_{t-1}(\boldsymbol{x}) := \mu_{t-1}(k^*;\boldsymbol{x})$ and $\sigma_{t-1}(\boldsymbol{x}) := \sigma_{t-1}(k^*;\boldsymbol{x})$ with $\bar{\sigma}^2 = 1+2/t$. We conclude that the base learner does not require knowledge of the true kernel for constructing confidence sets, as long as there is sufficient meta-data available. Theorem 4.3 quantifies this notion of sufficiency.

Case Study: Bayesian Optimization As an example application, we consider the classic Bayesian optimization problem, but in the case where $\mathcal{H}_{k^*}$ is unknown. This example illustrates how Theorem 5.1 may be used to prove guarantees for a decision-making algorithm, which uses the meta-learned kernel due to a lack of knowledge of k^*. We follow the setup and BO notation of Srinivas et al. (2010). The agent seeks to maximize an unknown reward function f, sequentially accessed as described in Equation (1). Their goal is to choose actions $\boldsymbol{x}_t$ which maximize the cumulative reward achieved over T time steps. This is equivalent to minimizing the cumulative regret $R_T = \sum_{t=1}^{T}[f(\boldsymbol{x}^*) - f(\boldsymbol{x}_t)]$, where $\boldsymbol{x}^*$ is a global maximum of f. Note that if $R_T/T \to 0$ as $T \to \infty$ then $\max_{1\leq t\leq T} f(\boldsymbol{x}_t) \to f(\boldsymbol{x}^*)$, i.e., the learner converges to the optimal value. We will refer to this property as *sublinearity* of the regret. In the spirit of the GP-UCB algorithm (Srinivas et al., 2010), we choose the next point by maximizing the upper confidence bound as determined by Theorem 5.1

$$\boldsymbol{x}_t = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \hat{\mu}_{t-1}(\boldsymbol{x}) + \nu_t\hat{\sigma}_{t-1}(\boldsymbol{x}) \qquad (11)$$

[3] The functions $\hat{\mu}_{t-1}$ and $\hat{\sigma}_{t-1}$ are the posterior mean and variance of $\mathrm{GP}(0,\hat{k})$, conditioned on H_{t-1}, with noise variance $\bar{\sigma}^2$.

where a suitable choice for ν_t is suggested in Corollary 5.2.

Corollary 5.2 (A Regret Bound with META-KEL). *Let $\delta \in (0,1)$. Suppose $f \in \mathcal{H}_{k^*}$ with $\|f\|_{k^*} \leq B$ and that values of f are observed with zero-mean sub-Gaussian noise of variance proxy σ^2. Then, with probability greater than $1-\delta$,* GP-UCB *used together with $\hat{k}$ satisfies*

$$R_T = \mathcal{O}\Bigg(\sqrt{\hat{d}T\log T}\Big(B\big(1+\epsilon(n,m)\big) \\ + \sqrt{\hat{d}\log T + \log 1/\delta}\Big)\Bigg)$$

provided that the exploration coefficient is set to

$$\begin{aligned}\nu_t =& B\big(1+\epsilon(n,m)/2c_1\big) \\ &+ \sigma\sqrt{\hat{d}\log\left(1+\bar{\sigma}^{-2}t/c_1\right)+2+2\log(1/\delta)}.\end{aligned}$$

The proof is straightforward. Conditioned on the event that $f \in \mathcal{H}_{\hat{k}}$, we may directly use the regret bound of Chowdhury & Gopalan (2017). Then, by Theorem 4.3, we calculate the probability of this event (Appendix C.1). The Corollary relies on knowledge of a bound B on $\|f\|_{k^*}$. However, using techniques of Berkenkamp et al. (2019) it is possible to adapt it even to *unknown* B at increased (but still sublinear) regret.

Corollary 5.2 shows that GP-UCB using the meta-learned kernel guarantees sublinear regret. We obtain a $\mathcal{O}(\hat{d}B\log T\sqrt{T})$ rate for the regret which is tight compared to the $\mathcal{O}(d^*B\log T\sqrt{T})$ rate satisfied by the oracle. It is insightful to compare this convergence result to a scenario where the hypothesis space is misspecified. For a reward function $f \notin \mathcal{H}_{\hat{k}}$, Bogunovic & Krause (2021) show that the learner will not converge to the global optimum, since the cumulative regret has a lower bound of linear order $\mathcal{O}(T\sqrt{\log T})$. Corollary 5.2 suggests that by using a sparse kernel we can potentially find the optimal policy faster compared to when the complex kernel k_{full} is used. Recall that $d = \sum_{j=1}^{p} d_j$, by Theorem 2 of Chowdhury & Gopalan (2017) the regret of GP-UCB used together with k_{full} is bounded by $\mathcal{O}(dpB\log T\sqrt{T})$, since $\|f\|_{k_{\text{full}}} = p\|f\|_{k^*}$. Therefore, using the meta-learned kernel improves the regret bound by a factor of $\hat{d}/(dp)$, implying that the solution may be found faster. The results of our experiments in Figure 4 support this argument.

Note that our approach to guarantee a sublinear regret for GP-UCB without oracle knowledge of k^* naturally generalizes to other sequential decision tasks. In particular, any theoretical result relying on RKHS confidence intervals with a *known* kernel can be immediately extended to use those of the meta-learned kernel.

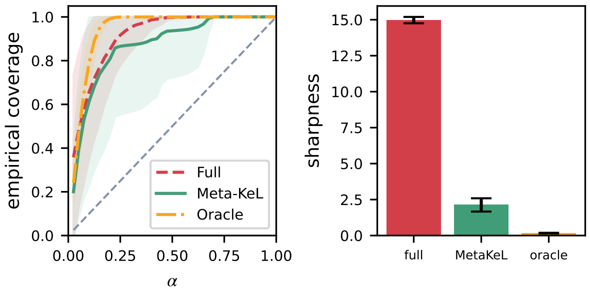


Figure 3. Calibration (left) and sharpness (right) experiment for confidence sets given 4 training samples. Averaged over 50 runs, $\hat{k}$ always gives tight valid confidence intervals.

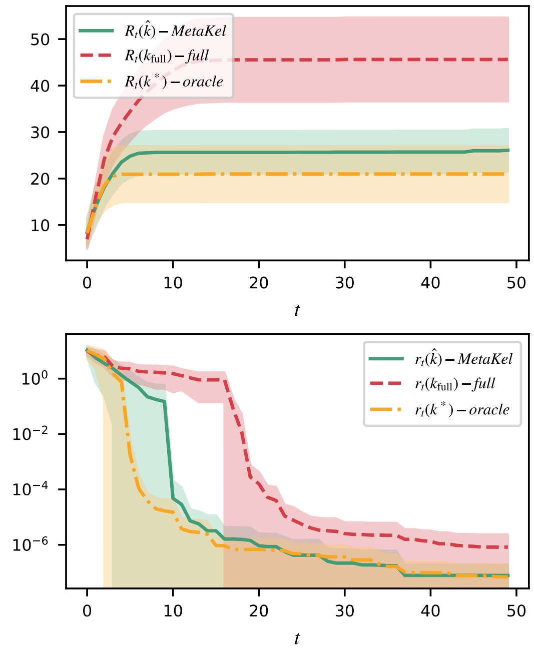


Figure 4. Simple and cumulative regret for GP-UCB. The algorithm converges at a slower pace when using k_{full}.

6. Experiments

In this section, we provide experiments to quantitatively illustrate our theoretical contribution.

Experiment Setup (1D) We create a synthetic dataset based on our data model, Equations (1) and (4). We first limit the domain to the 1-dimensional $\mathcal{X} = [-1,1]$ and use Legendre polynomials P_j as our features ϕ_j. The sequence $(P_j)_{j\geq 0}$ is a natural choice, since it provides an orthonormal basis for $L^2(\mathcal{X})$. Moreover, Legendre polynomials are eigenfunctions to dot-product kernels such as the Neural Tangent Kernel (Jacot et al., 2018). We let $k^*(x,x') = \sum_{j\in J_{k^*}} \eta_j^* P_j(x)P_j(x')$, where J_{k^*} is a random subset of $\{1,\cdots,p\}$. Each η_j^* is sampled independently from the standard uniform distribution and the vec-

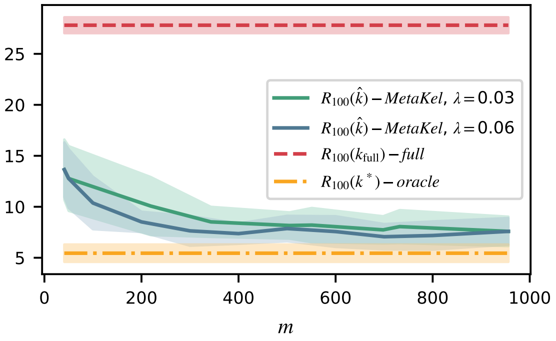


Figure 5. The regret of GP-UCB used with $\hat{k}$ approaches the oracle regret as the number of offline tasks increase.

tor η^* is then normalized. Across all experiments, we set $p = 20$ and $s = |J_{k^*}| = 5$. To sample the meta-data $\mathcal{D}_{n,m}$, we choose m independent random subsets of J_{k^*} and generate the functions f_s via Equation (6) where $\boldsymbol{\beta}^{*(j)}$ are drawn from an i.i.d. standard uniform distribution. We then scale the norm $\|f\|_{k^*}$ to $B = 10$. The data for a single task, i.e., $(x_{s,i}, y_{s,i})_{i \leq n}$, is then created by uniformly drawing i.i.d. samples from the domain $\mathcal{X}$ and evaluating f_s at those points. We add Gaussian noise with standard deviation $\sigma = 0.01$ to all data points. Figure 8 in the appendix shows how random f_s may look like. For all experiments we set $n = m = 50$ unless stated otherwise. To meta-learn $\hat{k}$, we solve the vectorized META-KEL problem (Eq. 8) over $\boldsymbol{\beta}^{(j)}$ with CELER, a fast solver for the group Lasso (Massias et al., 2018), and then set $\hat{\boldsymbol{\eta}}$ according to Proposition 4.1. We set $\lambda = 0.03$, such that it satisfies the condition of Theorem 4.3. As shown in Figure 10 in the appendix, the choice of λ has little effect on the performance of the algorithm.

Confidence Set Experiment We perform calibration and sharpness experiments to assess the meta-learned confidence sets (Gneiting et al., 2007). Figure 3 presents the result. To obtain an α-confidence interval for some $f(x)$ using a kernel k, we assume a $f \sim \mathrm{GP}(0, k)$ prior and calculate the α-quantile of the posterior after observing 4 noisy i.i.d. draws from the function. For each hypothesis, the y-axis of the left plot shows the empirical coverage of the confidence sets, i.e., the fraction of test points contained in the α-confidence intervals for varying levels α. In this plot, if a curve were to fall below the $x = y$ line, it would have implied insufficient coverage and hence over-confident sets. The plot on the right shows the posterior variance averaged across all test points. This quantity, referred to as *sharpness*, reflects the width of the confidence bands. Figure 3 demonstrates that the meta-learned confidence sets are well-calibrated for the entire range of confidence-levels and are tight relative to the true sets. In contrast, k_{full} yields conservative confidence sets, due to considering polynomials P_j that do not contribute to the construction of $f(x)$. We use 1000 test points for calculating the empirical averages. The plot shows the values averaged over 50 runs, where for each the kernel k^* and the data are generated from scratch.

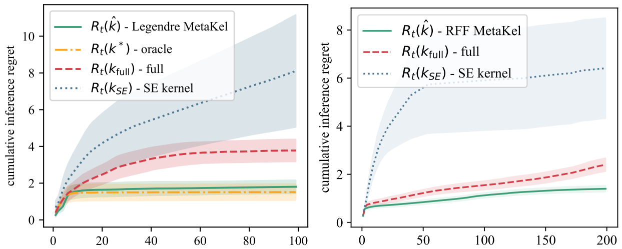


Figure 6. Cumulative regret of GP-UCB. A synthetic 2D BO (left), hyper-parameters tuning of GLMNET (right).

Regret Experiment We verify the performance of GP-UCB when used together with $\hat{k}$. We generate the random reward function f in a manner similar to f_s of the meta-data. The BO problem is simulated according to Equation (1), and the actions are selected via Equation (11). Figure 9 in the appendix shows how this algorithm samples the domain and how the confidence estimates shrink by observing new samples. Keeping the underlying random k^* fixed, we generate 100 random instances of the meta-learning and the BO problem. In Figure 4 we present the average regret and its standard deviation. In these plots, the simple regret of GP-UCB with a kernel k is labeled $r_t(k) = f(x^*) - \max_{\tau \leq t} f(x_\tau)$. Respectively, the cumulative inference regret is $R_t(k) = \sum_{\tau \leq t} f(x^*) - \max_x \mu_{\tau-1}(x)$. The algorithm converges to the optimum using all three kernels. The meta-learned kernel, however, improves upon using k_{full} and results in a performance competitive to when k^* is known by GP-UCB. This behavior empirically confirms Corollary 5.2.

Consistency Experiment From Corollary 5.2 we concluded that, as the size of the meta-data grows, the regret bound achieved via $\hat{k}$ converges to the oracle bound, i.e., the bound satisfied by the learner when it has knowledge of the true kernel. As Figure 5 shows, this consistency is also reflected in the empirical regret values. As we increase m, the number of offline tasks given to the meta-learner, the cumulative inference regret at $T = 100$ improves. It approaches the regret obtained by the oracle algorithm. The value of λ does not affect this convergence, as long as it satisfies Theorem 4.3. Similar to Figure 4, this plot is generated for a fixed random k^*, averaged over 50 random instances of the meta-learning and BO problem.

Regret Experiment for 2D domain We repeat the regret experiment, with synthetic data over the 2-dimensional domain $\mathcal{X} = [-1, 1]^2$. For $\boldsymbol{x} = (x_1, x_2) \in \mathcal{X}$, we define the Legendre feature map as $\boldsymbol{\phi}(\boldsymbol{x}) = (P_j(x_1)P_{p-j}(x_2))_{0 \leq j \leq p}$. This feature map is $(p+1)$-dimensional, and has terms of degree at most p. We use the polynomial $P_j(x_1)P_{p-j}(x_2)$ as the feature $\boldsymbol{\phi}_j(\boldsymbol{x})$ and create a synthetic dataset in a fashion identical to the 1-dimensional case, again setting $p = 20$ and $s = 5$. Figure 6 shows the 2-dimensional counterpart of Figure 4. Again, using $\hat{k}$ results in competitive performance to the oracle algorithm with knowledge of k^*. Here, we also use GP-UCB together with the infinite-dimensional Squared Exponential (SE) kernel. The regret curve shows that we do not benefit from choosing a complex kernel for solving an inherently low dimensional problem.

Efficient Hyper-parameter Tuning with META-KEL A common application of GP-UCB is in optimizing hyper-parameters of machine learning algorithms. In this setting, $\mathcal{X}$ is the algorithm's hyper-parameter space, and f represents the test performance of the algorithm. Evaluating each hyper-parameter configuration is costly, and thus the BO method has to be sample efficient. We consider the GLMNET algorithm (Friedman et al., 2010) and empirically demonstrate that by meta-learning the kernel, we gather knowledge from prior data, and in turn, tune the hyper-parameters of a new task more efficiently. Mainly, by running GP-UCB with $\hat{k}$ we tend to find the optimal configuration of hyper-parameters for an unseen learning task faster, compared to using data-independent kernels (Figure 6). The OpenML platform (Bischl et al., 2017) enables access to data from hyper-parameter tuning of GLMNET on 38 different classification tasks. We split these datasets into a meta-dataset with $m = 25$ and leave the rest as test tasks. For meta-learning the kernel, we use 500 Random Fourier Features (Rahimi et al., 2007) defined on a 2-dimensional domain, since the GLMNET algorithm has only two hyper-parameters. The remaining details of our experiment setup is given in Appendix D. Figure 6 shows the performance of GP-UCB on the test task. Utilizing $\hat{k}$ results in a sample-efficient GP-UCB that rapidly approaches an optimal choice of hyper-parameters. This is in contrast to running GP-UCB with the SE kernel, which takes about 50 iterations to find a good configuration.

7. Conclusion

We proposed META-KEL, a method for reliably learning kernel functions from offline meta-data. As a first, our approach yields provably valid meta-learned adaptive confidence sets, setting it apart from existing meta-learning approaches. Importantly, we showed that our meta-learned kernel produces tight and consistent confidence bounds for the target function, provided that enough meta-data is available. As an example application, we showed that GP-UCB still yields sublinear regret when using the meta-learned kernel, performing competitively in theory and experiments with the oracle that has access to the unknown kernel. We believe this result opens up avenues towards giving convergence guarantees for other sequential decision algorithms without oracle knowledge of the hypothesis space.

Acknowledgements

We thank Andreea Musat and Victor Armegioiu for fruitful discussions and their contributions to an earlier research path of this project. In addition, we thank Scott Sussex and Anastasia Makarova for their feedback on the draft of this paper. We appreciate Felix Schur and Adrian Müller's thorough feedback on the final manuscript. This research was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program grant agreement no. 815943. Jonas Rothfuss was supported by an Apple Scholars in AI/ML fellowship.

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, 2011.

Bach, F., Jenatton, R., Mairal, J., Obozinski, G., et al. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 2012.

Bach, F. R. Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, 2008.

Bach, F. R., Lanckriet, G. R., and Jordan, M. I. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, 2004.

Bastani, H. and Bayati, M. Online decision making with high-dimensional covariates. *Operations Research*, 2020.

Basu, S., Kveton, B., Zaheer, M., and Szepesvári, C. No regrets for learning the prior in bandits. *Advances in Neural Information Processing Systems*, 34, 2021.

Berkenkamp, F., Turchetta, M., Schoellig, A., and Krause, A. Safe model-based reinforcement learning with stability guarantees. *Advances in neural information processing systems*, 2017.

Berkenkamp, F., Schoellig, A. P., and Krause, A. No-regret bayesian optimization with unknown hyperparameters. *Journal of Machine Learning Research*, 2019.

Bickel, P. J., Ritov, Y., and Tsybakov, A. B. Simultaneous analysis of lasso and dantzig selector. *The Annals of statistics*, 2009.

Bischl, B., Casalicchio, G., Feurer, M., Hutter, F., Lang, M., Mantovani, R., Rijn, J. N., and Vanschoren, J. Openml benchmarking suites and the openml100. *arXiv preprint arXiv:1708.03731*, 2017.

Bogunovic, I. and Krause, A. Misspecified Gaussian process bandit optimization. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Boutilier, C., Hsu, C.-w., Kveton, B., Mladenov, M., Szepesvari, C., and Zaheer, M. Differentiable meta-learning of bandit policies. In *Advances in Neural Information Processing Systems*, 2020.

Boyd, S., Boyd, S. P., and Vandenberghe, L. *Convex optimization*. Cambridge University Press, 2004.

Bühlmann, P. and Van De Geer, S. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

Bunea, F., Lederer, J., and She, Y. The group square-root lasso: Theoretical properties and fast algorithms. *IEEE Transactions on Information Theory*, 2013.

Cavalier, L., Golubev, G., Picard, D., and Tsybakov, A. Oracle inequalities for inverse problems. *The Annals of Statistics*, 2002.

Cella, L. and Pontil, M. Multi-task and meta-learning with sparse linear bandits. In *Uncertainty in Artificial Intelligence*. PMLR, 2021.

Cella, L., Lazaric, A., and Pontil, M. Meta-learning with stochastic linear bandits. In *International Conference on Machine Learning*. PMLR, 2020.

Chowdhury, S. R. and Gopalan, A. On kernelized multi-armed bandits. In *International Conference on Machine Learning*. PMLR, 2017.

Cristianini, N., Kandola, J., Elisseeff, A., and Shawe-Taylor, J. On kernel target alignment. In *Innovations in machine learning*. Springer, 2006.

Curi, S., Berkenkamp, F., and Krause, A. Efficient model-based reinforcement learning through optimistic policy search and planning. *Advances in Neural Information Processing Systems*, 2020.

Evgeniou, T. and Pontil, M. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004.

Friedman, J., Hastie, T., and Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 2010.

Gneiting, T., Balabdaoui, F., and Raftery, A. E. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2007.

Gönen, M. and Alpaydın, E. Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 2011.

Hao, B., Lattimore, T., and Wang, M. High-dimensional sparse linear bandits. *Advances in Neural Information Processing Systems*, 2020.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 2018.

Javanmard, A. and Montanari, A. Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 2014.

Kakade, S., Krishnamurthy, A., Lowrey, K., Ohnishi, M., and Sun, W. Information theoretic regret bounds for online nonlinear control. *Advances in Neural Information Processing Systems*, 2020.

Kim, G.-S. and Paik, M. C. Doubly-robust lasso bandit. *Advances in Neural Information Processing Systems*, 2019.

Kloft, M., Brefeld, U., Sonnenburg, S., and Zien, A. Lp-norm multiple kernel learning. *The Journal of Machine Learning Research*, 2011.

Koltchinskii, V. and Yuan, M. Sparse recovery in large ensembles of kernel machines. In *Proceedings of Conference on Learning Theory*, 2008.

Kühn, D., Probst, P., Thomas, J., and Bischl, B. Automatic exploration of machine learning experiments on openml. *arXiv preprint arXiv:1806.10961*, 2018.

Kveton, B., Mladenov, M., Hsu, C.-W., Zaheer, M., Szepesvari, C., and Boutilier, C. Meta-learning bandit policies by gradient ascent. *arXiv preprint arXiv:2006.05094*, 2020.

Liu, H. and Zhang, J. Estimation consistency of the group lasso and its applications. In *Artificial Intelligence and Statistics*. PMLR, 2009.

Lounici, K., Pontil, M., Van De Geer, S., and Tsybakov, A. B. Oracle inequalities and optimal inference under group sparsity. *The annals of statistics*, 2011.

Massias, M., Gramfort, A., and Salmon, J. Celer: a fast solver for the lasso with dual extrapolation. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Ong, C. S., Smola, A. J., and Williamson, R. C. Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, 2005.

Perrone, V., Jenatton, R., Seeger, M. W., and Archambeau, C. Scalable Hyperparameter Transfer Learning. In *Advances in Neural Information Processing Systems*, 2008.

Rahimi, A., Recht, B., et al. Random features for large-scale kernel machines. In *NIPS*, 2007.

Rothfuss, J., Fortuin, V., Josifoski, M., and Krause, A. Pacoh: Bayes-optimal meta-learning with pac-guarantees. In *International Conference on Machine Learning*. PMLR, 2021a.

Rothfuss, J., Heyn, D., Chen, J., and Krause, A. Meta-learning reliable priors in the function space. In *Advances in Neural Information Processing Systems*, 2021b.

Russo, D. and Van Roy, B. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 2014.

Sessa, P. G., Bogunovic, I., Kamgarpour, M., and Krause, A. Learning to play sequential games versus unknown opponents. *Advances in Neural Information Processing Systems*, 2020.

Simchowitz, M., Tosh, C., Krishnamurthy, A., Hsu, D. J., Lykouris, T., Dudik, M., and Schapire, R. E. Bayesian decision-making under misspecified priors with applications to meta-learning. *Advances in Neural Information Processing Systems*, 2021.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010.

Vakili, S., Khezeli, K., and Picheny, V. On information gain and regret bounds in gaussian process bandits. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021.

Van de Geer, S., Bühlmann, P., and Zhou, S. The adaptive and the thresholded lasso for potentially misspecified models (and a lower bound for the lasso). *Electronic Journal of Statistics*, 2011.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press, 2018.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*. Cambridge University Press, 2019.

Wang, X., Wei, M., and Yao, T. Minimax concave penalized multi-armed bandit model with high-dimensional covariates. In *International Conference on Machine Learning*. PMLR, 2018a.

Wang, Z. and de Freitas, N. Theoretical analysis of bayesian optimisation with unknown gaussian process hyper-parameters. *arXiv preprint arXiv:1406.7758*, 2014.

Wang, Z., Li, C., Jegelka, S., and Kohli, P. Batched high-dimensional bayesian optimization via structural kernel learning. In *International Conference on Machine Learning*. PMLR, 2017.

Wang, Z., Kim, B., and Kaelbling, L. P. Regret bounds for meta bayesian optimization with an unknown gaussian process prior. *arXiv preprint arXiv:1811.09558*, 2018b.

Wynne, G., Briol, F.-X., and Girolami, M. Convergence guarantees for gaussian process means with misspecified likelihoods and smoothness. *Journal of Machine Learning Research*, 2021.

Zhao, P. and Yu, B. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 2006.

Zhou, D., Li, L., and Gu, Q. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*. PMLR, 2020.

Meta-Learning Hypothesis Spaces for Sequential Decision-making: Supplementary Material

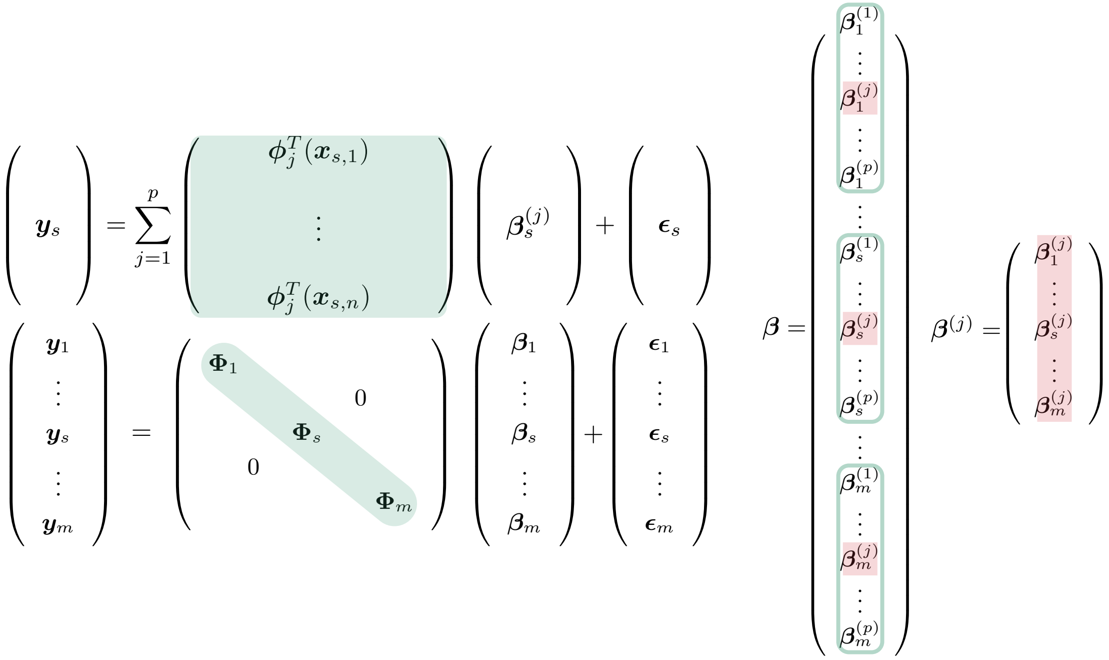
A. Details of the main Result

symbol	description
k^*	true (unknown) kernel function
$\hat{k}$	meta-learned kernel
p	total number of candidate base kernels
k_j	base kernels that construct k^* and $\hat{k}$, $1 \leq j \leq p$
η^*_j	coefficient of k_j in construction of k^*, i.e., $k^*(\cdot,\cdot) = \sum_{j=1}^p \eta^*_j k_j(\cdot,\cdot)$
$\hat{\eta}_j$	coefficient of k_j in construction of $\hat{k}$, i.e., $\hat{k}(\cdot,\cdot) = \sum_{j=1}^p \hat{\eta}_j k_j(\cdot,\cdot)$
$\boldsymbol{\eta}^*$	the vector $\left(\eta^*_1, \cdots, \eta^*_j, \cdots, \eta^*_p\right) \in \mathbb{R}^p$
$\hat{\boldsymbol{\eta}}$	the vector $(\hat{\eta}_1, \cdots, \hat{\eta}_j, \cdots, \hat{\eta}_p) \in \mathbb{R}^p$
J_{k^*}	the set $\{1 \leq j \leq p,\ \text{s.t.}\, \eta^*_j \neq 0\}$
$J_{\hat{k}}$	the set $\{1 \leq j \leq p,\ \text{s.t.}\, \hat{\eta}_j \neq 0\}$
$\boldsymbol{\phi}_j(\cdot)$	feature map for k_j, i.e., $k_j(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}_j^T(\boldsymbol{x})\boldsymbol{\phi}_j(\boldsymbol{x}')$
$\boldsymbol{\phi}(\cdot)$	$\left(\sqrt{\eta^*_1}\boldsymbol{\phi}_1^T(\boldsymbol{x}), \cdots, \sqrt{\eta^*_p}\boldsymbol{\phi}_p^T(\boldsymbol{x})\right)^T$, i.e., feature map for k^*
d_0	dimension of the input domain, $\mathcal{X} \subset \mathbb{R}^{d_0}$
d_j	dimension of a feature map $\boldsymbol{\phi}_j(\boldsymbol{x}) \in \mathbb{R}^{d_j}$
$d_{\max}$	$\max_{1 \leq j \leq p} d_j$
d	$\sum_{j=1}^p d_j$
d^*	$\sum_{j \in J_{k^*}} d_j$
$\hat{d}$	$\sum_{j \in J_{\hat{k}}} d_j$
n	number of samples in each task s in the meta-data
m	number of tasks in the meta-data
f_s	target function from task s
$\boldsymbol{\beta}^*_s$	true coefficients vector for task s: $f_s(\boldsymbol{x}) = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{\beta}^*_s$, in $\mathbb{R}^d$
$\hat{\boldsymbol{\beta}}_s$	estimate of $\boldsymbol{\beta}^*_s$
$\boldsymbol{\beta}^{*\,(j)}_s$	sub-vector of $\boldsymbol{\beta}^*_s$ corresponding to kernel k_j, in $\mathbb{R}^{d_j}$
$\hat{\boldsymbol{\beta}}_s{}^{(j)}$	estimate of $\boldsymbol{\beta}^{*\,(j)}_s$
$\boldsymbol{\beta}^*$	$(\boldsymbol{\beta}^{*\,T}_1, \cdots, \boldsymbol{\beta}^{*}_m{}^T)^T$, in $\mathbb{R}^{md}$
$\hat{\boldsymbol{\beta}}$	estimate of $\boldsymbol{\beta}^*$
$\boldsymbol{\beta}^{*\,(j)}$	sub-vector of $\boldsymbol{\beta}^*$ corresponding to kernel k_j, in $\mathbb{R}^{md_j}$
$\hat{\boldsymbol{\beta}}^{(j)}$	estimate of $\boldsymbol{\beta}^{*\,(j)}$

Table 1. Notation Guide

A.1. RKHS Refresher

Here we present a compact reminder of RKHS basics for the sake of completeness and clarifying our notation. We work in a finite-dimensional regime which can also be described by a euclidean vector space. Nevertheless, we use the RKHS notation as it gives a powerful framework and hides away the vector algebra. This section is mainly based on Wainwright (2019). For a positive semi-definite kernel function k over some set $\mathcal{X} \times \mathcal{X}$, the corresponding unique Reproducing Kernel Hilbert



$\boldsymbol{y}_s = \sum_{j=1}^{p} \begin{pmatrix} \boldsymbol{\phi}_j^T(\boldsymbol{x}_{s,1}) \\ \vdots \\ \boldsymbol{\phi}_j^T(\boldsymbol{x}_{s,n}) \end{pmatrix} \boldsymbol{\beta}_s^{(j)} + \boldsymbol{\epsilon}_s$
$\begin{pmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_s \\ \vdots \\ \boldsymbol{y}_m \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Phi}_1 & & 0 \\ & \boldsymbol{\Phi}_s & \\ 0 & & \boldsymbol{\Phi}_m \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_s \\ \vdots \\ \boldsymbol{\beta}_m \end{pmatrix} + \begin{pmatrix} \boldsymbol{\epsilon}_1 \\ \vdots \\ \boldsymbol{\epsilon}_s \\ \vdots \\ \boldsymbol{\epsilon}_m \end{pmatrix}$
$\boldsymbol{\beta} = (\boldsymbol{\beta}_1^{(1)}, \dots, \boldsymbol{\beta}_1^{(j)}, \dots, \boldsymbol{\beta}_1^{(p)}, \dots, \boldsymbol{\beta}_s^{(1)}, \dots, \boldsymbol{\beta}_s^{(j)}, \dots, \boldsymbol{\beta}_s^{(p)}, \dots, \boldsymbol{\beta}_m^{(1)}, \dots, \boldsymbol{\beta}_m^{(j)}, \dots, \boldsymbol{\beta}_m^{(p)})$
$\boldsymbol{\beta}^{(j)} = (\boldsymbol{\beta}_1^{(j)}, \dots, \boldsymbol{\beta}_s^{(j)}, \dots, \boldsymbol{\beta}_m^{(j)})$

Figure 7. Visual guide for the Group Lasso formulation. The red shade shows the group of coefficients which correspond to the effect of kernel k_j. The green shade demonstrates how features from each task come together on the diagonal of the multi-task feature matrix. The coefficients that belong to one task are group together with a green rectangle.

Space can be constructed as,

$$\mathcal{H}_k = \Big\{ f : \mathcal{X} \to \mathbb{R} \,|\, f(\cdot) = \sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}_i, \cdot),\, n \in \mathbb{N},\, (\boldsymbol{x}_i)_{i=1}^n \in \mathcal{X}, \boldsymbol{\alpha} \in \mathbb{R}^n \Big\},$$

equipped with the dot product, $\langle f, \bar{f} \rangle_k = \sum_{i,j} \alpha_i \bar{\alpha}_j k(x_i, \bar{x}_j)$. We limit $\mathcal{X}$ to compact sets, and only consider Mercer kernels, i.e., continuous kernel functions that satisfy the Hilbert-Schmidt condition,

$$\int_{\mathcal{X}\times\mathcal{X}} k^2(\boldsymbol{x}, \boldsymbol{x}') d\mu(\boldsymbol{x}) d\mu(\boldsymbol{x}') < \infty$$

where μ is a non-negative measure over $\mathcal{X}$. Mercer's theorem states that under these assumptions, the kernel operator has a sequence of orthonormal eigenfunctions $(\phi_r)_{r\geq 1}$and non-negative eigenvalues $(\eta_r)_{r\geq 1}$, defined as follows

$$\int_{\mathcal{X}} k(\boldsymbol{x}, \boldsymbol{x}') \phi_r(\boldsymbol{x}') d\mu(\boldsymbol{x}') = \eta_r \phi_r(\boldsymbol{x}). \tag{A.1}$$

Moreover, k can be written as their linear combination,

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_r \eta_r \phi_r(\boldsymbol{x}) \phi_r(\boldsymbol{x}').$$

Or as a non-negative combination of base kernels, $k_r(\boldsymbol{x}, \boldsymbol{x}') = \phi_r(\boldsymbol{x}) \phi_r(\boldsymbol{x}')$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_r \eta_r k_r(\boldsymbol{x}, \boldsymbol{x}').$$

It immediately follows that the unique RKHS corresponding to k takes the form,

$$\mathcal{H}_k = \left\{ f : \mathcal{X} \to \mathbb{R} \,|\, f(\cdot) = \sum_{r\geq 1} \beta_r \phi_r(\cdot),\; \sum_{r:\eta_r \neq 0} \frac{\beta_r^2}{\eta_r} < \infty \right\}$$

and the inner product the following form,

$$\langle f, g\rangle_k = \sum_{r:\eta_r \neq 0} \frac{\langle f, \phi_r\rangle_2 \langle g, \phi_r\rangle_2}{\eta_r},$$

where $\langle \cdot, \cdot\rangle_2$ denotes the inner product in the $L^2(\mathcal{X})$. It is then implied that $\|f\|_k = \sum_{r:\eta_r\neq 0} \beta_r^2/\eta_r$. Lastly, we define $\boldsymbol{\phi}(\boldsymbol{x}) = \left(\sqrt{\eta_r}\phi_r(\boldsymbol{x})\right)_{r\geq 1}$ to be the feature map corresponding to k. In this paper we refer to the number of non-zero eigen-values as the dimension of the kernel. Note that under this convention, most kernel functions used in practice, e.g. RBF kernel or the Matérn family, are infinite-dimensional.

A.2. Proof of Proposition 4.1

By Equation (6) we may write a parametric equivalent of Problem (7) in terms of the feature maps $\boldsymbol{\phi}_j$,

$$\begin{aligned}\min_{\substack{0\leq \boldsymbol{\eta},\\ \forall s: \boldsymbol{\beta}_s}} \frac{1}{m}\sum_{s=1}^m \left[\frac{1}{n}\sum_{i=1}^n \left(y_{s,i} - \sum_{j=1}^p \sqrt{\eta_j}\boldsymbol{\phi}_j^T(\boldsymbol{x}_{s,i})\boldsymbol{\beta}_s^{(j)}\right)^2\right] \\ + \frac{\lambda}{2}\sum_{s=1}^m\sum_{j=1}^p \left\|\boldsymbol{\beta}_s^{(j)}\right\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\eta}\|_1.\end{aligned} \tag{A.2}$$

This problem is jointly convex in $\boldsymbol{\eta}$ and $(\boldsymbol{\beta}_s)_{s\leq m}$, and has an optimal solution (Kloft et al., 2011). Renaming the variable $\boldsymbol{\beta}_s^{(j)} \leftarrow \boldsymbol{\beta}_s^{(j)}/\sqrt{\eta_j}$ gives the following equivalent problem,

$$\min_{\substack{0\leq \boldsymbol{\eta},\\ \forall s: \boldsymbol{\beta}_s}} \frac{1}{m}\sum_{s=1}^m \left[\frac{1}{n}\sum_{i=1}^n \left(y_{s,i} - \sum_{j=1}^p \boldsymbol{\phi}_j^T(\boldsymbol{x}_{s,i})\boldsymbol{\beta}_s^{(j)}\right)^2\right] + \frac{\lambda}{2}\sum_{j=1}^p \frac{\sum_{s=1}^m \left\|\boldsymbol{\beta}_s^{(j)}\right\|_2^2}{\eta_j} + \frac{\lambda}{2}\|\boldsymbol{\eta}\|_1.$$

There is no constraint connecting the two variables, and renaming $\boldsymbol{\beta}_s^{(j)}$ does not effect the optimization problem with respect to η. Therefore, $\hat{\boldsymbol{\eta}}$ is also an optima for this problem. Let $(\hat{\boldsymbol{\eta}}, \hat{\boldsymbol{\beta}})$ denote the solution to the problem above. We show that $\hat{\eta}$ has a closed form expression in terms of $\hat{\boldsymbol{\beta}}$. This observation allows us to reduce the joint optimization problem to an equivalent problem which is only over $(\boldsymbol{\beta}_s)$. We use the *η-trick* introduced in Bach et al. (2004). The authors observe that for any two scalar variables w and v,

$$|w| = \min_{v\geq 0} \frac{w^2}{2v} + \frac{v}{2},$$

and $\hat{v} = |w|$. Applying this trick with $w = \left\|\boldsymbol{\beta}^{(j)}\right\|_2$ and $v = \eta_j$ for all $j \leq p$ gives

$$\min_{\boldsymbol{\eta}\geq 0} \frac{\lambda}{2}\sum_{j=1}^p \frac{\left\|\boldsymbol{\beta}^{(j)}\right\|_2^2}{\eta_j} + \frac{\lambda}{2}\|\boldsymbol{\eta}\|_1 = \lambda\sum_{j=1}^p \left\|\boldsymbol{\beta}^{(j)}\right\|_2,$$

which results the following equivalent problem

$$\min_{\forall s: \boldsymbol{\beta}_s} \frac{1}{m}\sum_{s=1}^m \left[\frac{1}{n}\sum_{i=1}^n \left(y_{s,i} - \sum_{j=1}^p \boldsymbol{\phi}_j^T(\boldsymbol{x}_{s,i})\boldsymbol{\beta}_s^{(j)}\right)^2\right] + \lambda\sum_{j=1}^p \sqrt{\sum_{s=1}^m \left\|\boldsymbol{\beta}_s^{(j)}\right\|_2^2}. \tag{A.3}$$

Note that By definition of $\boldsymbol{\beta}^{(j)}$, the second term satisfies $\sum_{j=1}^p \sqrt{\sum_{s=1}^m \left\|\boldsymbol{\beta}_s^{(j)}\right\|_2^2} = \left\|\boldsymbol{\beta}^{(j)}\right\|_2$. Finally, by simply using the vectorized notation we get Equation (8), concluding the proof.

B. Proof of Statements in Section 4

For the first three lemmas in this section we follow the technique in Lounici et al. (2011) and occasionally use classical ideas established in Bühlmann & Van De Geer (2011).

Notations and naming conventions When X is a matrix, $\|X\|_2$ and $\|X\|_{\mathrm{F}}$ denote its spectral and Frobenius norm, respectively. Consider the multi-task coefficients vector $\boldsymbol{\beta} \in \mathbb{R}^{md}$, and the sub-vector $\boldsymbol{\beta}^{(j)} \in \mathbb{R}^{md_j}$ which denotes the coefficients corresponding to kernel k_j. Through out this proof, we use the convention "group j" to refer to the set of indices of $\boldsymbol{\beta}$ which indicate $\boldsymbol{\beta}^{(j)}$. Similarly, we let $\boldsymbol{\Phi}^{(j)}$ denote the $mn \times md_j$ sub-matrix which only has the features coming from group j. Lastly, let $\boldsymbol{\Psi} := \boldsymbol{\Phi}^T\boldsymbol{\Phi}/mn$, then $\boldsymbol{\Psi}^{(j)} = (\boldsymbol{\Phi}^{(j)})^T\boldsymbol{\Phi}^{(j)}/mn$ indicates the $md_j \times md_j$ submatrix that is caused by group j.

B.1. Proof of Theorem 4.3

Recall the vectorized formulation of the META-KEL loss,

$$\mathcal{L}(\boldsymbol{\beta}) = \frac{1}{mn}\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\beta}\|_2^2 + \lambda\sum_{j=1}^{p}\left\|\boldsymbol{\beta}^{(j)}\right\|_2.$$

Let $\boldsymbol{\varepsilon}_s = (\varepsilon_{s,i})_{i\leq n}$ denote error for task s and $\boldsymbol{\varepsilon} \in \mathbb{R}^{mn}$ the stacked multi-task error vector. Using $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, we may decompose the loss into two deterministic and random parts. The term $2\boldsymbol{\varepsilon}^T\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)/mn$ is the random one and we will refer to as the empirical process. The first typical step in bounding the estimation error of Lasso estimators, is showing that the empirical process, which comes from the noise in observing values of $\boldsymbol{y}$, does not play a drastic role. More formally, let $A_j = \left\{\left\|(\boldsymbol{\Phi}^T\boldsymbol{\varepsilon})^{(j)}\right\|_2/mn \leq \lambda/4\right\}$ denote that the event that the image of noise affecting the feature space of $\boldsymbol{\phi}_j$, is dominated by the regularization term of $\boldsymbol{\beta}^{(j)}$. In Lemma B.1 we show that $\cap_{j=1}^{p}A_j$ happens with high probability, if λ is set properly.

Lemma B.1 (Regularization term dominates the empirical process)**.** *Set* $0 < \delta < 1$. *Consider the random event* $A = \cap_{j=1}^{p}A_j$. *Then* A *happens with probability greater than* $1-\delta$, *if*

$$\lambda \geq \frac{4\sigma}{\sqrt{mn}}\sqrt{1 + \frac{2}{m}\left(\log(2p/\delta) + \sqrt{md_{\max}\log(2p/\delta)}\right)}.$$

where $d_{\max} = \max_{1\leq j\leq p} d_j$.

We now show that if the empirical process is controlled by regularization, i.e. if λ is set to be large enough, then $\hat{\boldsymbol{\beta}} = \min\mathcal{L}(\boldsymbol{\beta})$ has favorable properties.

Lemma B.2 (Conditional properties of $\hat{\boldsymbol{\beta}}$)**.** *Assume that event* A *happens. Then for any solution* $\hat{\boldsymbol{\beta}}$ *of problem 8 and all* $\boldsymbol{\beta} \in \mathbb{R}^{md}$, *the following hold:*

$$\begin{aligned}\frac{1}{mn}\left\|\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\right\|_2^2 + \frac{\lambda}{2}\sum_{j=1}^{p}\left\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{*(j)}\right\|_2 &\leq \frac{1}{mn}\|\boldsymbol{\Phi}(\boldsymbol{\beta} - \boldsymbol{\beta}^*)\|_2^2 \\ &+ 2\lambda\sum_{j:\boldsymbol{\beta}^{(j)}\neq 0}\min\left(\left\|\boldsymbol{\beta}^{(j)}\right\|_2, \left\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{(j)}\right\|\right) \qquad \text{(B.1)}\end{aligned}$$

$$\left|\left\{j : \hat{\boldsymbol{\beta}}^{(j)} \neq 0\right\}\right| \leq \frac{16}{(mn\lambda)^2}\left\|\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\right\|_2^2 \qquad \text{(B.2)}$$

Note that by the meta data-generating model (Section 3), $J_{k^*} = \{j : \boldsymbol{\beta}^{*(j)} \neq 0\}$.

Lemma B.3 (Complete Variable Screening)**.** *Assume* $|J_{k^*}| \leq s$ *and set* $0 < \delta \leq 1$. *Define*

$$\epsilon(n,m) = \frac{32\sigma s}{\kappa^2\sqrt{mn}}\sqrt{1 + \frac{2}{m}\left(\log(2p/\delta) + \sqrt{md_{\max}\log(2p/\delta)}\right)}$$

Under assumption 4.2 with $\kappa = \kappa(s)$, *if* λ *is chosen according to lemma B.1, then with probability greater than* $1-\delta$

$$\max_{j\in J_{k^*}}\left\|\hat{\boldsymbol{\beta}}^{(j)} - (\boldsymbol{\beta}^*)^{(j)}\right\|_2 \leq \epsilon(n,m) \qquad \text{(B.3)}$$

and if in addition $\min_{j\in J_{k^*}}\left\|\boldsymbol{\beta}^{*(j)}\right\|_2 \geq c_1$, *then with the same probability for all* $j \in J_{k^*}$

$$\left|\left\|\hat{\boldsymbol{\beta}}^{(j)}\right\|_2 - c_1\right| \leq \epsilon(n,m). \qquad \text{(B.4)}$$

We now turn to the first claim made in Theorem 4.3, and prove that $\mathcal{H}_{k^*} \subseteq \mathcal{H}_{\hat{k}}$. Since $\hat{\eta}_j \geq 0$ and k_i are Mercer, then $\hat{k}$ is also Mercer and corresponds to an RKHS which we have been referring to as $\mathcal{H}_{\hat{k}}$. Consider the RKHS $\mathcal{H}_{k^*}$, since $|J_{k^*}|$ and each d_js are finite,

$$\mathcal{H}_{k^*} = \left\{ f : f(\cdot) = \sum_{j \in J_{k^*}} \sqrt{\eta_j^*} \boldsymbol{\beta}_j^T \boldsymbol{\phi}_j(\cdot),\, \boldsymbol{\beta}_j \in \mathbb{R}^d,\, \|\boldsymbol{\beta}_j\| < \infty \right\}$$

Therefore, $f \in \mathcal{H}_{k^*}$ if and only if it is in the finite span of $\boldsymbol{\phi}$, defined as

$$\mathrm{FinSpan}\left(\{\boldsymbol{\phi}_j : j \in J_{k^*}\}\right) = \left\{ f : f(\cdot) = \sum_{j \in J_{k^*}} \boldsymbol{\beta}_j^T \boldsymbol{\phi}_j(\cdot),\, \boldsymbol{\beta}_j \in \mathbb{R}^d,\, \|\boldsymbol{\beta}_j\| < \infty \right\}$$

Lemma B.3 states that $\hat{\boldsymbol{\beta}}^{(j)} \geq c_1 - \epsilon(n,m)$ with probability greater than $1 - \delta$ for $j \in J_{k^*}$. Therefore, for any $j \in J_{k^*}$, we get $\hat{\eta}_j \geq 1 - \epsilon(n,m)/c_1$, since we had set $\hat{\eta}_j = \left\|\hat{\boldsymbol{\beta}}^{(j)}\right\|/c_1$. If $c_1 > \epsilon(n,m)$, then $\hat{\eta}_j > 0$ and $j \in J_{\hat{k}}$, implying $J_{k^*} \subset J_{\hat{k}}$. Hence, under the assumptions of the theorem, with probability greater than $1 - \delta$,

$$\mathcal{H}_{k^*} = \mathrm{FinSpan}\left(\{\boldsymbol{\phi}_j : j \in J_{k^*}\}\right) \overset{\text{w.h.p.}}{\subset} \mathrm{FinSpan}\left(\{\boldsymbol{\phi}_j : j \in J_{\hat{k}}\}\right) = \mathcal{H}_{\hat{k}}.$$

The next lemma bounds the $\hat{k}$-norm of functions contained in $\mathcal{H}_{k^*}$, concluding the proof for Theorem 4.3.

Lemma B.4 (Bounding the $\hat{k}$-norm)**.** *Set $0 < \delta \leq 1$ and choose λ according to Lemma B.1 and define $\epsilon(n,m)$ according to Lemma B.3. If $|J_{k^*}| \leq s$ and Assumption 4.2 holds with $\kappa = \kappa(s)$, then under Condition 3.1, for all $f \in \mathcal{H}_{k^*}$ with $\|f\|_{k^*} \leq B$,*

$$\|f\|_{\hat{k}} \leq \left(1 + \frac{\epsilon(n,m)}{2c_1} + o\left(\epsilon(n,m)\right)\right) \tag{B.5}$$

B.2. Proof of Proposition 4.4

Under assumptions of the proposition, event A happens with probability greater than $1 - \delta$. From Equation (B.2) in Lemma B.2,

$$\left|J_{\hat{k}}\right| \leq \frac{16}{(mn\lambda)^2} \left\|\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\right\|_2^2$$

and by Equation (B.10),

$$\frac{1}{\sqrt{mn}} \left\|\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\right\|_2 \leq \frac{2\lambda\sqrt{s}}{\kappa}$$

which gives

$$|J_{\hat{k}}| \leq \frac{64s}{mn\kappa^2(s)}$$

Therefore, if $mn = \mathcal{O}(s/p)$ then $\left|J_{\hat{k}}\right| \leq p = |J_{k_{\text{full}}}|$ with probability greater than $1 - \delta$. and via similar argument as given in the proof of theorem 4.3,

$$\mathcal{H}_{\hat{k}} = \mathrm{FinSpan}\left(\{\boldsymbol{\phi}_j : j \in J_{\hat{k}}\}\right) \overset{\text{w.h.p.}}{\subset} \mathrm{FinSpan}\left(\{\boldsymbol{\phi}_j : j \in J_{k_{\text{full}}}\}\right) = \mathcal{H}_{k_{\text{full}}}.$$

B.3. Proof of Lemmas used in Section B.1

This section presents the proofs to the helper lemmas introduced before.

Proof of Lemma B.1. This proof follows a similar treatment of the empirical process to Lemma 3.1 Lounici et al. (2011). Since $\varepsilon_{i,s}$ are i.i.d. zero-mean sub-gaussian variables, we observe that

$$\mathbb{P}(A_j) = \mathbb{P}\left(\left\{\frac{1}{(mn)^2}\boldsymbol{\varepsilon}^T \boldsymbol{\Phi}^{(j)} (\boldsymbol{\Phi}^{(j)})^T \boldsymbol{\varepsilon} \leq \frac{\lambda^2}{16}\right\}\right) = \mathbb{P}\left(\left\{\frac{\sum_{i=1}^{mn} v_i(z_i^2 - 1)}{\sqrt{2}\|\boldsymbol{v}\|} \leq \alpha\right\}\right)$$

where z_i are i.i.d. sub-gaussian variables with variance proxy 1, v_i denote the eigenvalues of $\boldsymbol{\Phi}^{(j)}(\boldsymbol{\Phi}^{(j)})^T/mn$ and $\boldsymbol{v}$ is the vector of these eigenvalues. Lastly,

$$\alpha = \frac{mn\lambda^2/(16\sigma^2) - \mathrm{Tr}(\boldsymbol{\Psi}^{(j)})}{\sqrt{2}\left\|\boldsymbol{\Psi}^{(j)}\right\|_{\mathrm{F}}}$$

From Equation 27 Cavalier et al. (2002) yields the following inequality,

$$\mathbb{P}(A_j^c) = \mathbb{P}\left(\left|\frac{\sum_{i=1}^{mn}(z_i^2-1)v_i}{\sqrt{2}\|\boldsymbol{v}\|_2}\right| > \alpha\right) \le 2\exp\left(-\frac{\alpha^2}{2(1+\sqrt{2}\alpha\|\boldsymbol{v}\|_\infty/\|\boldsymbol{v}\|_2)}\right)$$

We choose λ such that the right hand side is bounded by δ/p. By definition of $\boldsymbol{v}$ we have $\|\boldsymbol{v}\|_\infty/\|\boldsymbol{v}\|_2 = \left\|\Psi^{(j)}\right\|_2/\left\|\Psi^{(j)}\right\|_{\mathrm{F}}$. Then, for A_j^c to happen with probability smaller than δ/p,

$$\lambda \ge \frac{4\sigma}{\sqrt{mn}}\sqrt{\mathrm{Tr}(\boldsymbol{\Psi}^{(j)}) + 2\left\|\boldsymbol{\Psi}^{(j)}\right\|_2\left(2\log(2p/\delta) + \sqrt{md_j\log(2p/\delta)}\right)}.$$

Then by union bound, A happens with probability greater than $1-\delta$ if

$$\lambda \ge \max_j \frac{4\sigma}{\sqrt{mn}}\sqrt{\mathrm{Tr}(\boldsymbol{\Psi}^{(j)}) + 2\left\|\boldsymbol{\Psi}^{(j)}\right\|_2\left(2\log(2p/\delta) + \sqrt{md_j\log(2p/\delta)}\right)}.$$

Since the base kernels are normalized we may bound the norm and trace of $\boldsymbol{\Psi}^{(j)}$.

$$\mathrm{Tr}(\boldsymbol{\Psi}^{(j)}) = \frac{1}{mn}\sum_{s=1}^{m}\mathrm{Tr}\left((\boldsymbol{\Phi}_s{}^{(j)})^T\boldsymbol{\Phi}_s{}^{(j)}\right) = \frac{1}{mn}\sum_{s=1}^{m}\sum_{i=1}^{n}\boldsymbol{\phi}_j^T(\boldsymbol{x}_{s,i})\boldsymbol{\phi}_j(\boldsymbol{x}_{s,i}) = \frac{1}{mn}\sum_{s=1}^{m}\sum_{i=1}^{n}k_j(\boldsymbol{x}_{s,i},\boldsymbol{x}_{s,i}) \le 1$$

Similarly, $\left\|\boldsymbol{\Psi}^{(j)}\right\|_2 \le \frac{1}{mn}\max_s\sum_{i=1}^n k_j(\boldsymbol{x}_{s,i},\boldsymbol{x}_{s,i}) \le 1/m$, and thereby concluding the proof. □

***Proof of Lemma B.2*.** For proving this lemma we are essentially only using Cauchy-Schwarz and the Triangle inequality, together with the KKT optimality condition for $\mathcal{L}$. For any $\boldsymbol{\beta}$, since $\hat{\boldsymbol{\beta}}$ is the minimizer of $\mathcal{L}$ and due to the data generating model (Eq. 4), we have

$$\frac{1}{mn}\left\|\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^*)\right\|_2^2 \le \frac{1}{mn}\|\boldsymbol{\Phi}(\boldsymbol{\beta}-\boldsymbol{\beta}^*)\|_2^2 + \frac{2}{mn}\boldsymbol{\varepsilon}^T\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}) + \lambda\sum_{j=1}^{p}\left(\left\|\boldsymbol{\beta}^{(j)}\right\|_2 - \left\|\hat{\boldsymbol{\beta}}^{(j)}\right\|_2\right).$$

By Cauchy-Schwarz and the assumption that A happens,

$$\boldsymbol{\varepsilon}^T\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}) \le \sum_{j=1}^{p}\left\|(\boldsymbol{\Phi}^T\boldsymbol{\varepsilon})^{(j)}\right\|_2\left\|\hat{\boldsymbol{\beta}}^{(j)}-\boldsymbol{\beta}^{(j)}\right\|_2 \le \frac{mn\lambda}{4}\sum_{j=1}^{p}\left\|\hat{\boldsymbol{\beta}}^{(j)}-\boldsymbol{\beta}^{(j)}\right\|_2,$$

and thereby,

$$\frac{1}{mn}\left\|\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^*)\right\|_2^2 + \frac{\lambda}{2}\sum_{j=1}^{p}\left\|\hat{\boldsymbol{\beta}}^{(j)}-\boldsymbol{\beta}^{(j)}\right\|_2 \le \frac{1}{mn}\|\boldsymbol{\Phi}(\boldsymbol{\beta}-\boldsymbol{\beta}^*)\|_2^2 + \lambda\sum_{j=1}^{p}\left(\left\|\hat{\boldsymbol{\beta}}^{(j)}-\boldsymbol{\beta}^{(j)}\right\|_2 + \left\|\boldsymbol{\beta}^{(j)}\right\|_2 - \left\|\hat{\boldsymbol{\beta}}^{(j)}\right\|_2\right)$$

which gives Inequality B.1. By the KKT optimality conditions for convex losses (Boyd et al., 2004), $\hat{\boldsymbol{\beta}}$ is a minimizer of $\mathcal{L}$, if and only if $0 \in \partial\mathcal{L}(\hat{\boldsymbol{\beta}})$, where $\partial\mathcal{L}(\hat{\boldsymbol{\beta}})$ denotes the sub-gradient of the loss evaluated at $\hat{\boldsymbol{\beta}}$. Therefore $\hat{\boldsymbol{\beta}}$ satisfies

$$\frac{2}{mn}\left(\boldsymbol{\Phi}^T(\boldsymbol{y}-\boldsymbol{\Phi}\hat{\boldsymbol{\beta}})\right)^{(j)} = \frac{\lambda\hat{\boldsymbol{\beta}}^{(j)}}{\left\|\hat{\boldsymbol{\beta}}^{(j)}\right\|}, \quad \text{if } \hat{\boldsymbol{\beta}}^{(j)} \neq 0 \tag{B.6}$$

$$\frac{2}{mn}\left\|\left(\boldsymbol{\Phi}^T(\boldsymbol{y}-\boldsymbol{\Phi}\hat{\boldsymbol{\beta}})\right)^{(j)}\right\|_2 \le \lambda, \quad \text{if } \hat{\boldsymbol{\beta}}^{(j)} = 0. \tag{B.7}$$

As for Inequality B.2, conditioned on event A, by Equation (4) together with the KKT condition B.6, we obtain that for all $1 \leq j \leq p$ where $\hat{\boldsymbol{\beta}}^{(j)} \neq 0$,

$$\frac{1}{mn}\left\|(\boldsymbol{\Phi}^T\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*))^{(j)}\right\|_2 \geq \frac{\lambda}{4}.$$

Following the analysis of Lounici et al. we conclude,

$$\begin{aligned}\left|\left\{j : \hat{\boldsymbol{\beta}}^{(j)} \neq 0\right\}\right| &\leq \frac{16}{(mn\lambda)^2}\sum_{j:\, \hat{\boldsymbol{\beta}} \neq 0}\left\|(\boldsymbol{\Phi}^T\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*))^{(j)}\right\|_2^2 \\ &\leq \frac{16}{(mn\lambda)^2}\sum_{j=1}^{p}\left\|(\boldsymbol{\Phi}^T\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*))^{(j)}\right\|_2^2 \\ &\leq \frac{16}{(mn\lambda)^2}\left\|(\boldsymbol{\Phi}^T\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*))\right\|_2^2 \\ &\leq \frac{16}{(mn\lambda)^2}\left\|(\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*))\right\|_2^2.\end{aligned}$$

Since the kernels k_j are normalized by 1 and $\|\boldsymbol{\Phi}\|_2 \leq \max_j k_j(\boldsymbol{x}, \boldsymbol{x}) \leq 1$. □

Proof of Lemma B.3. Let $\boldsymbol{\beta}_{J_{k^*}} = (\boldsymbol{\beta}^{(j)})_{j \in J_{k^*}}$ denote the sub coefficient vector that corresponds to all the active groups. Due to Equation (B.1) with $\boldsymbol{\beta} = \boldsymbol{\beta}^*$, conditioned on event A we have

$$\frac{1}{mn}\left\|\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\right\|_2^2 \leq 2\lambda \sum_{j \in J_{k^*}}\left\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{*(j)}\right\| \leq 2\lambda\sqrt{\sum_{j \in J_{k^*}} s\left\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{*(j)}\right\|_2^2} = 2\lambda\sqrt{s}\left\|\hat{\boldsymbol{\beta}}_{J_{k^*}} - \boldsymbol{\beta}^*{}_{J_{k^*}}\right\|_2 \tag{B.8}$$

where the second inequality follows from Cauchy-Schwarz together with the Lemma's assumption $|J_{k^*}| \leq s$. By again using Equation (B.1) we get $\sum_{j=1}^{p}\left\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{*(j)}\right\|_2 \leq 4\sum_{j \in J_{k^*}}\left\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{*(j)}\right\|_2$, and thereby $\sum_{j \notin J_{k^*}}^{p}\left\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{*(j)}\right\|_2 \leq 3\sum_{j \in J_{k^*}}\left\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{*(j)}\right\|_2$. Using assumption 4.2, this inequality indicates that

$$\left\|\hat{\boldsymbol{\beta}}_{J_{k^*}} - \boldsymbol{\beta}^*{}_{J_{k^*}}\right\|_2 \leq \frac{1}{\kappa\sqrt{mn}}\left\|\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\right\|_2, \tag{B.9}$$

which together with Equation (B.8) gives,

$$\frac{1}{\sqrt{mn}}\left\|\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\right\|_2 \leq \frac{2\lambda\sqrt{s}}{\kappa} \tag{B.10}$$

The next chain of inequalities proves the first statement of the Lemma (Equation (B.3)). For all $j \in J_{k^*}$,

$$\begin{aligned}\left\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{*(j)}\right\|_2 &\leq \sum_{j=1}^{p}\left\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{*(j)}\right\|_2 \\ &\overset{\text{B.1}}{\leq} 4\sum_{j \in J}\left\|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}^{*(j)}\right\|_2 \\ &\overset{\text{B.8}}{\leq} 4\sqrt{s}\left\|\hat{\boldsymbol{\beta}}_{J_{k^*}} - \boldsymbol{\beta}^*{}_{J_{k^*}}\right\|_2 \\ &\overset{\text{B.9}}{\leq} \frac{4\sqrt{s}}{\kappa\sqrt{mn}}\left\|\boldsymbol{\Phi}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\right\|_2 \\ &\overset{\text{B.10}}{\leq} \frac{8\lambda s}{\kappa^2}\end{aligned}$$

Note that the analysis here is carried out conditional on event A. From Lemma B.1, we have that A happens with probability greater than $1 - \delta$ if λ is set according to the statement of the Lemma. Recall that from the Beta-min condition (Cond. 3.1) we have $\left\|\boldsymbol{\beta}^{*(j)}\right\| > c_1$, which together with Equation (B.3) concludes the proof. □

Proof of Lemma B.4. Let $f \in \mathcal{H}_{k^*}$, with $\|f\|_k^2 \le B^2$. Then by construction of $\mathcal{H}_{k^*}$ (see Appendix A.1)

$$f(\boldsymbol{x}) = \sum_{j=1}^{p} \sqrt{\eta_j^*}\,\boldsymbol{\phi}_j^T(\boldsymbol{x})(\boldsymbol{\beta})^{(j)}, \quad \|f\|_k^2 = \sum_{j:\eta_j^*\neq 0} \left\|(\boldsymbol{\beta})^{(j)}\right\|_2^2 = \|\boldsymbol{\beta}\|_2^2.$$

We calculate the $\hat{k}$ norm of functions that lie in $\mathcal{H}_{k^*}$. Let $I = \{1 \le j \le p : \eta_j^* \neq 0, \hat\eta_j \neq 0\}$.

$$\|f\|_{\hat k}^2 = \sum_{j:\hat\eta_j\neq0}\sum_{r=1}^{d_j}\frac{(\langle f,\phi_{j,r}\rangle_2)^2}{\hat\eta_j} = \sum_{j:\hat\eta_j\neq 0}\sum_{r=1}^{d_j}\frac{\left(\sqrt{\eta_j^*}\boldsymbol{\beta_r}^{(j)}\right)^2}{\hat\eta_j} = \sum_{j\in I}\frac{\eta_j^*}{\hat\eta_j}\left\|\boldsymbol\beta^{(j)}\right\|_2^2 = \sum_{j\in I}\frac{\eta^*_j}{\hat\eta_j}\left\|\boldsymbol\beta^{(j)}\right\|_2^2$$

where $\phi_{j,r}$ denotes the r-th feature in the feature map $\boldsymbol{\phi}_j$, similarly $\boldsymbol{\beta_r}^{(j)}$ the r-th element in vector $\boldsymbol{\beta}^{(j)}$, and $\langle\cdot,\cdot\rangle_2$ is the inner product in the $L^2(\mathcal{X})$ space. By applying Cauchy-Schwarz we get

$$\|f\|_{\hat k}^2 \le \sqrt{\sum_{j\in I}(\frac{\eta_j^*}{\hat\eta_j})^2}\cdot\sqrt{\sum_{j\in I}\|\boldsymbol\beta^{(j)}\|_2^4}.$$

Consider the vector $\boldsymbol{v} = (\left\|\boldsymbol\beta^{(j)}\right\|_2^2)_{j\in I}$, we observe that $\|\boldsymbol{v}\|_2 = \sqrt{\sum_{j\in I}\left\|\boldsymbol\beta^{(j)}\right\|_2^4}$ and $\|\boldsymbol{v}\|_1 \le B^2$. Since $\|\cdot\|_2 \le \|\cdot\|_1$ and due to the assumption $\|\boldsymbol\eta^*\|_1 \le 1$, we obtain

$$\|f\|_{\hat k}^2 \le B^2\sum_{j\in I}\frac{\eta_j^*}{\hat\eta_j} \le B^2\|\boldsymbol\eta^*\|_1\max_{j\in I}\frac{1}{\hat\eta_j} \le B^2\max_{j\in I}\frac{1}{\hat\eta_j}. \tag{B.11}$$

It remains to bound $\max_{j\in I}\hat\eta_j^{-1}$. We set $\hat{\boldsymbol\eta}_{\boldsymbol j} = \left\|\hat{\boldsymbol\beta}^{(j)}\right\|_2/(c1)$ and under conditions of the theorem, Lemma B.3 states that $\hat{\boldsymbol\eta}_j \ge 1-\epsilon(n,m)/c_1$, for all $j \in J_{k^*}$. Then for members of $I \subset J_{k^*}$,

$$\frac{1}{\hat\eta_j} \le \frac{1}{1-\epsilon(n,m)/c_1} \le \left(1+\frac{\epsilon(n,m)}{c_1}+o\left(\epsilon(n,m)\right)\right),$$

which implies the following for the norm bound

$$\|f\|_{\hat k} \le B\left(1+\frac{\epsilon(n,m)}{2c_1}+o\left(\epsilon(n,m)\right)\right) \tag{B.12}$$

with probability greater than $1-\delta$. □

C. Proof of Statements in Section 5

The following lemma presents a confidence bound for when the learner has oracle knowledge of the true kernel. This lemma plays an integral role in for the proofs in this section.

Lemma C.1 (Theorem 2 Chowdhury & Gopalan (2017) for $\hat k$)**.** *Let $f \in \mathcal{H}_{\hat k}$ for some kernel $\hat k$ with a $\hat k$-norm bounded by $\hat B$. Then with probability greater than $1-\bar\delta$, for all $\boldsymbol{x}\in\mathcal{X}$ and $t \ge 1$,*

$$|\hat\mu_{t-1}(\boldsymbol{x}) - f(\boldsymbol{x})| \le \hat\sigma_{t-1}(\boldsymbol{x})\left(\hat B + \sigma\sqrt{2(\hat\gamma_{t-1}+1+\log(1/\bar\delta))}\right) \tag{C.1}$$

where $\hat\mu_{t-1}$ and $\hat\sigma_{t-1}$ are as defined in Equation 3 with $\bar\sigma = 1+2/T$.

We skip the proof for this lemma as it is given in Chowdhury & Gopalan (2017), with the same notation. For the kernel $\hat k$ we define the *maximum information gain* after $t-1$ observations as

$$\hat\gamma_{t-1} := \max_{[\boldsymbol{x}_\tau]_{\tau\le t}}\frac{1}{2}\log\det(\boldsymbol{I}+\bar\sigma^{-2}\hat{\boldsymbol{K}}_{t-1})$$

This parameter quantifies the speed at which we learn about f, when using the kernel $\hat k$. Note that γ_{t-1} is independent of any specific realization of H_{t-1}. It only depends on the choice of kernel, the input domain, and the noise variance. The next lemma bounds this parameter.

Lemma C.2 (Information Gain Bound)**.** *The maximum information gain for $\hat{k}$ after observing t samples satisfies,*

$$\hat{\gamma}_t \leq \frac{\hat{d}}{2}\log\big(1+\frac{\bar{\sigma}^{-2}t}{c_1}\big) = \mathcal{O}(\hat{d}\log t/c_1)$$

where $\hat{d} = \sum_{j\in J_{\hat{k}}} d_j \leq d$.

We now have the main tools for proving Theorem 5.1.

Proof of Theorem 5.1*.* Assume that $f \in \mathcal{H}_{k^*}$, and that $\|f\|_{k^*} \leq B$. Then by Theorem 4.3 $f \in \mathcal{H}_{\hat{k}}$, with probability greater than $1-\delta$. Define $\hat{B}$ as

$$\hat{B} = B\left(1+\frac{\epsilon(n,m)}{2c_1} + o\left(\epsilon(n,m)\right)\right), \tag{C.2}$$

then by Lemma B.4, $\|f\|_{\hat{k}} \leq \hat{B}$ with probability greater than $1-\delta$. We first condition on the event that $f \in \mathcal{H}_{\hat{k}}$ and $\|f\|_{\hat{k}} \leq \hat{B}$. Then Lemma C.1 gives the following confidence interval,

$$\mathbb{P}\left(|\hat{\mu}_{t-1}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \hat{\sigma}_{t-1}(\boldsymbol{x})\left(\hat{B} + \sigma\sqrt{2(\hat{\gamma}_{t-1}+1+\log(1/\delta))}\right)|f\in\mathcal{H}_{\hat{k}}, \|f\|_{\hat{k}} \leq \hat{B}\right) \geq 1-\bar{\delta} \tag{C.3}$$

We now remove the conditional event. Let $C_t(\boldsymbol{x}) := [\hat{\mu}(\boldsymbol{x}) - \hat{\sigma}(\boldsymbol{x}), \hat{\mu}(\boldsymbol{x}) - \hat{\sigma}(\boldsymbol{x})]$. By the chain rule,

$$\mathbb{P}\left(f(\boldsymbol{x}) \in C_t(\boldsymbol{x})\right) \geq \mathbb{P}\left(f(\boldsymbol{x}) \in C_t(\boldsymbol{x})|f\in\mathcal{H}_{\hat{k}}\right)\cdot\mathbb{P}\left(f\in\mathcal{H}_{\hat{k}}\right) \geq 1-\delta-\bar{\delta}$$

Renaming $\delta+\bar{\delta}$ to δ for simplicity, we conclude that with probability greater than $1-\delta$

$$|\hat{\mu}_{t-1}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \hat{\sigma}_{t-1}(\boldsymbol{x})\left(B\left(1+\frac{\epsilon(n,m)}{2c_1}+o\left(\epsilon(n,m)\right)\right) + \sigma\sqrt{2(\hat{\gamma}_{t-1}+1+\log(1/\delta))}\right).$$

Lastly, Lemma C.2 gives an upper bound for $\hat{\gamma}_{t-1}$ which completes the proof. □

Proof of Lemma C.2*.* For this proof, we follow a similar technique as in Vakili et al. (2021). Recall that $\hat{k}(\boldsymbol{x},\boldsymbol{x}') = \sum_{j\in J_{\hat{k}}} \hat{\eta}_j \boldsymbol{\phi}_j^T(\boldsymbol{x})\boldsymbol{\phi}_j(\boldsymbol{x}')$, where $J_{\hat{k}} = \{1 \leq j \leq p : \hat{\eta}_j \neq 0\}$. Define $\hat{d}$ the effective dimension of kernels k_j that correspond to this index set, $\hat{d} = \sum_{j\in J_{\hat{k}}} d_j$. Now consider any arbitrary sequence of inputs $(\boldsymbol{x}_\tau)_{\tau=1}^t$ and let $\hat{\Phi}_t = \left(\hat{\boldsymbol{\phi}}(\boldsymbol{x}_1),\cdots,\hat{\boldsymbol{\phi}}(\boldsymbol{x}_t)\right) \in \mathbb{R}^{t\times\hat{d}}$, with $\hat{\boldsymbol{\phi}}(\boldsymbol{x}) = \left(\boldsymbol{\phi}_j(\boldsymbol{x})\right)_{j\in J_{\hat{k}}}$. Define the $\hat{d}\times\hat{d}$ matrix $\Lambda = \operatorname{diag}\left((\hat{\eta}_j)_{j\in J_{\hat{k}}}\right)$ as the diagonal matrix containing the eigenfunctions of $\hat{k}$. We have $K_t = \hat{\Phi}_t\Lambda\hat{\Phi}_t^T$. Let $H_t = \Lambda^{1/2}\hat{\Phi}^T\hat{\Phi}\Lambda^{1/2}$, by the Weinstein-Aronszajn identity,

$$\begin{aligned}\frac{1}{2}\log\det(\boldsymbol{I}+\bar{\sigma}^{-2}K_t) &= \frac{1}{2}\log\det(\boldsymbol{I}+\bar{\sigma}^{-2}H_t)\\ &\leq \frac{1}{2}\hat{d}\log\left(\operatorname{tr}(\boldsymbol{I}+\bar{\sigma}^{-2}H_t)/\hat{d}\right)\end{aligned}$$

For positive definite matrices $\boldsymbol{P}\in\mathbb{R}^{n\times n}$, we have $\log\det\boldsymbol{P} \leq n\log\operatorname{tr}(\boldsymbol{P}/n)$. The inequality follows from $\boldsymbol{I}+\bar{\sigma}^{-2}H_t$ being positive definite, since $\hat{\eta}_j \geq 0$. We may write,

$$\begin{aligned}\frac{1}{2}\log\det(\boldsymbol{I}+\bar{\sigma}^{-2}K_t) &\leq \frac{1}{2}\hat{d}\log\big(1+\frac{\bar{\sigma}^{-2}}{\hat{d}}\operatorname{tr}(\Lambda^{1/2}\hat{\Phi}^T\hat{\Phi}\Lambda^{1/2})\big)\\ &\leq \frac{1}{2}\hat{d}\log\big(1+\frac{\bar{\sigma}^{-2}}{\hat{d}}\sum_{\tau=1}^t\operatorname{tr}(\Lambda^{1/2}\hat{\boldsymbol{\phi}}^T(\boldsymbol{x}_\tau)\hat{\boldsymbol{\phi}}(\boldsymbol{x}_\tau)\Lambda^{1/2})\big)\\ &\leq \frac{1}{2}\hat{d}\log\big(1+\frac{\bar{\sigma}^{-2}}{\hat{d}}\sum_{\tau=1}^t||\hat{\boldsymbol{\phi}}(\boldsymbol{x}_\tau)\Lambda^{1/2}||_2^2\big)\\ &\leq \frac{1}{2}\hat{d}\log\big(1+\frac{\bar{\sigma}^{-2}}{\hat{d}}\sum_{\tau=1}^t\sum_{j\in J_{\hat{k}}}\hat{\eta}_j\|\boldsymbol{\phi}_j(\boldsymbol{x}_\tau)\|_2^2\big)\\ &\leq \frac{1}{2}\hat{d}\log\big(1+\frac{\bar{\sigma}^{-2}t}{c_1}\max_j\left\|\boldsymbol{\beta}^{(j)}\right\|_2\big)\end{aligned}$$

The next to last inequality holds since $k_j(\boldsymbol{x}, \boldsymbol{x}) = \|\boldsymbol{\phi}_j(\boldsymbol{x})\|_2^2$ is normalized to one and $\hat{\eta}_j = \left\|\hat{\boldsymbol{\beta}}_{(j)}\right\|_2 / c_1$. The inequality above holds for any sequence $(\boldsymbol{x}_\tau)_{\tau \leq t}$,

$$\hat{\gamma}_t = \mathcal{O}\left(\hat{d} \log(t/c_1)\right)$$

□

C.1. Proof of Corollary 5.2

Similar to the previous section, we take advantage of a classic regret bound for the oracle learner and then apply it to our setting.

Lemma C.3 (Theorem 3 Chowdhury & Gopalan (2017), for $\hat{k}$)**.** *Set $\delta \in (0, 1)$. If $f \in \mathcal{H}_{\hat{k}}$ with $\|f\|_{\hat{k}} \leq \hat{B}$, then with probability $1 - \delta$,* GP-UCB *satisfies,*

$$R_T = \mathcal{O}\left(\hat{B}\sqrt{T\hat{\gamma}_T} + \sqrt{T\hat{\gamma}_T(\hat{\gamma}_T + \log 1/\delta)}\right)$$

Proof of Corollary 5.2. Conditioned on the event that $f \in \mathcal{H}_{\hat{k}}$ and $\|f\|_{\hat{k}} \leq \hat{B}$, Lemma C.3 states that

$$\mathbb{P}\left(R_T = \mathcal{O}\left(\hat{B}\sqrt{T\hat{\gamma}_T} + \sqrt{T\hat{\gamma}_T(\gamma_T + \log 1/\delta)}\right) | f \in \mathcal{H}_{\hat{k}}, \|f\|_{\hat{k}} \leq \hat{B}\right) \geq 1 - \bar{\delta}$$

Then by Lemma B.4 and Theorem 4.3, with probability greater than $1 - \delta - \bar{\delta}$

$$R_T = \mathcal{O}\left(\hat{B}\sqrt{T\hat{\gamma}_T} + \sqrt{T\hat{\gamma}_T(\hat{\gamma}_T + \log 1/\delta)}\right)$$

where $\hat{B}$ is set according to Equation (C.2). Plugging in Lemma C.2 to bound the information gain and changing the variable name $\delta + \bar{\delta}$ to δ and concludes the proof.

□

D. Experiments

Regret Experiment on Hyper-Parameter Tuning data This experiment is based on Rothfuss et al. (2021b, Appendix B.1.2), we repeat some of the details for completeness. We consider the use case of hyper-parameter tuning for machine learning algorithms. In particular, we consider Generalized linear models with elastic NET regularization (GLMNET) (Friedman et al., 2010) for this purpose, which has two hyper-parameters `lambda` and `alpha`. Following previous work (e.g., Perrone et al., 2008), we replace the costly training and evaluation step by a cheap table lookup based on a large number of hyper-parameter evaluations (Kühn et al., 2018) on 38 classification datasets from the OpenML platform (Bischl et al., 2017). The hyper-parameter evaluations are available under a Creative Commons BY 4.0 license and can be downloaded here[4]. In effect, $\mathcal{X}$ is a finite set, corresponding to 10000-30000 random evaluations hyper-parameter evaluations per dataset and machine learning algorithm. Since the sampling is quite dense, for the purpose of empirically evaluating the meta-learned models towards BO, this finite domain can be treated like a continuous domain. All datasets correspond to binary classification. The target function we aim to optimize is the area under the ROC curve (AUROC) on a test split of the respective dataset. Since (Kühn et al., 2018) sample `lambda` in the log-space, we transform it via `lambda` $\leftarrow \log_2($`lambda`$)/10$ such that we can expect a reasonably good performance of a Vanilla GP-UCB with SE kernel. We randomly split the available tasks (i.e. train/test evaluations on a specific dataset) into a set of meta-train and meta-test tasks. In the following, we list the corresponding OpenML dataset identifiers:

- meta-train tasks: 3, 1036, 1038, 1043, 1046, 151, 1176, 1049, 1050, 31, 1570, 37, 4134, 1063, 1067, 44, 1068, 50, 1461, 1462
- test tasks: 335, 1489, 1486, 1494, 1504, 1120, 1510, 1479, 1480, 333, 1485, 1487, 334

The plots in Figure 6 show average cumulative regret for 13 test tasks, and 10 runs with different random seeds for each.

[4] https://doi.org/10.6084/m9.figshare.5882230.v2

Supplementary Figures Figure 8 illustrates a few samples of the random functions that we optimize over in the experiments. The functions are constructed using the Legendre basis, as explained in the main text. Figure 9 gives an example of a BO problem where we use IGP-UCB together with the meta-learned kernel to find the minimum of a function with as few samples as possible. As the function estimate improves, the confidence sets rapidly shrink and the learner only samples points close to the minimum. Figure 10 demonstrates that the choice of λ for the META-KEL loss does not have a severe effect on the regret of IGP-UCB. This is only the case if λ satisfies the condition of Theorem 4.3.

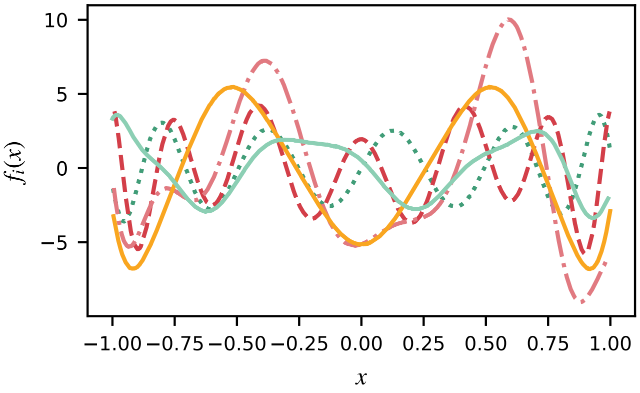


Figure 8. Examples of possible functions f_s for the meta-dataset.

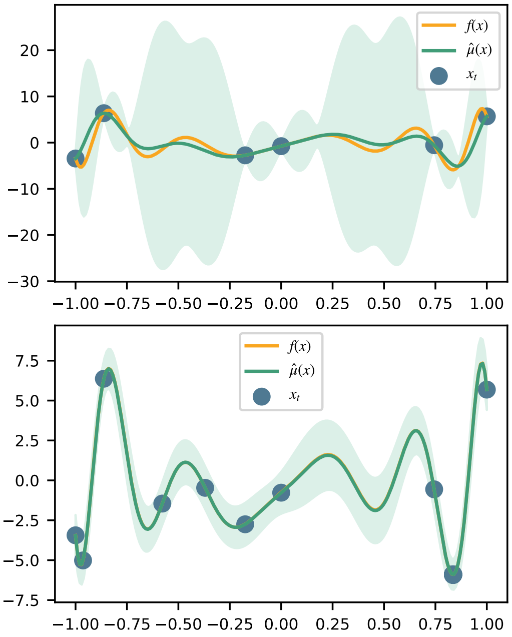


Figure 9. BO (minimization) with META-KEL. Upper plot shows the state at $t = 5$ and the lower plot at $t = 55$.

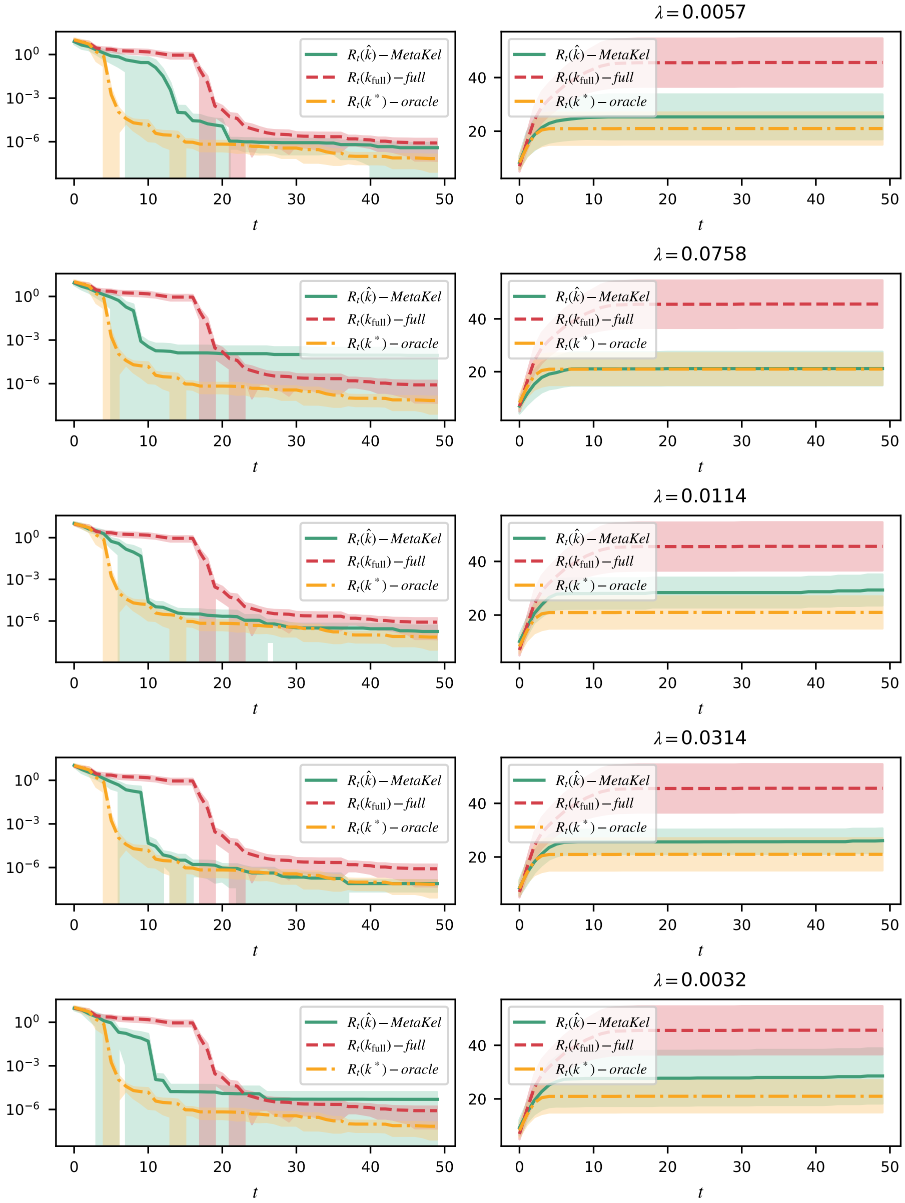


Figure 10. For $m = n = 50$ and $p = 20$, Theorem 4.3 requires that $\lambda > 0.001$ for recovery to happen with probability greater than $1 - \delta = 0.9$. For λ that satisfies this condition, the particular choice of its value does not effect performance severely.