# No-regret Algorithms for Multi-task Bayesian Optimization

**Sayak Ray Chowdhury**
Department of ECE
Indian Institute of Science
Bengaluru, India

**Aditya Gopalan**
Department of ECE
Indian Institute of Science
Bengaluru, India

## Abstract

We consider multi-objective optimization (MOO) of an unknown vector-valued function in the non-parametric Bayesian optimization (BO) setting. Our aim is to maximize the expected cumulative utility of all objectives, as expressed by a given prior over a set of scalarization functions. Most existing BO algorithms do not model the fact that the multiple objectives, or equivalently, tasks can share similarities, and even the few that do lack rigorous, finite-time regret guarantees that capture explicitly inter-task structure. In this work, we address this problem by modelling inter-task dependencies using a multi-task kernel and develop two novel BO algorithms based on scalarization of the objectives. Our algorithms employ vector-valued kernel regression as a stepping stone and belong to the upper confidence bound class of algorithms. Under a smoothness assumption that the unknown vector-valued function is an element of the reproducing kernel Hilbert space associated with the multi-task kernel, we derive worst-case regret bounds for our algorithms that explicitly capture the similarities between tasks. We numerically benchmark our algorithms on both synthetic and real-life MOO problems, and show the advantages offered by learning with multi-task kernels.

## 1 INTRODUCTION

Bayesian optimization (Frazier, 2018; Archetti and Candelieri, 2019) is a popular online learning approach for optimizing a black-box function with expensive, noisy evaluations, having been extensively applied in various applications such as hyper-parameter tuning (Snoek et al., 2012), sensor selection (Garnett et al., 2010), synthetic gene design (Gonzalez et al., 2015), etc. In many practical scenarios, one is required to optimize *multiple* objectives together, and moreover, these objectives can be conflicting in nature. For example, consider drug discovery, where each function evaluation is a costly laboratory experiment and its output is a measurement of both the potency and side-effects of a candidate drug (Paria et al., 2020). These two objectives are typically conflicting in nature, since one would like to maximize the potency of drug while also keeping its side-effects to a minimum. Other examples include trade-offs such as bias and variance, accuracy and calibration (Guo et al., 2017), accuracy and fairness (Zliobaite, 2015) etc. These problems can be framed as that of optimizing a vector-valued function $f = (f_1, \ldots, f_n)$, where each of its components is a real-valued function and corresponds to a particular objective or task.

The traditional goal in online MOO is a *search* goal – find the the set of *Pareto optimal* points using as few interactions with the environment as possible, where intuitively a point is Pareto optimal if there is no way to improve on all objectives simultaneously (Knowles, 2006). This is however, quite different than the corresponding *optimization* goal, where the learner seeks to maximize the total *utility* earned from all its decisions or equivalently, minimize the regret or shortfall in total utility compared to that of an optimal decision. This goal is relevant in many practical online MOO settings in which every decision that is taken carries utility or value. For example, consider the selection of different preventive strategies against an emerging epidemic under conflicting objectives like minimizing the infection rate while sustaining the economic growth (Roijers et al., 2017). In such cases, there is no separate budget or time devoted to purely exploring the unknown environment; rather, we need to use our interactions with the environment efficiently, as we care about the utility accrued during learning and hence, exploration and exploitation must be carefully balanced. Since one often cannot optimize all $f_i$'s simultaneously, what the optimal point is depends on the preferences of end user regarding the trade-offs between the different objectives. The preferences are typically expressed in terms of a scalarisation function (Roijers et al., 2013). *Random scalarizations*,

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. 

in particular, have been shown to be flexible enough to also model user preferences in capturing the whole or a part of the *Pareto front* (Paria et al., 2020).

Most multi-objective BO approaches maintain $n$ different Gaussian processes or, in short GPs (Gramacy, 2020), one for each task or objective $f_i$ (Zuluaga et al., 2013; Hernández-Lobato et al., 2016). However, in general, the tasks share some underlying structure, and cannot be treated as unrelated objects. By making use of this structure, one might benefit significantly by learning the tasks simultaneously as opposed to learning them independently. For example, consider predicting consumer preferences simultaneously based on their past history (Evgeniou et al., 2005). Each task is to learn the preference of a particular consumer, and the tasks are related since people with similar tastes tend to buy similar items. Other examples include simultaneous estimation of many related indicators in economic forecasting (Greene, 2003), predicting tumour behaviour from multiple related diseases (Rifkin et al., 2003) etc. However, assuming similarities in a set of tasks and blindly learning them together can be detrimental (Caruana, 1997). Hence, it is important to have a model that will benefit the learning in case of related tasks and will not hurt performance when the tasks are unrelated. This can be achieved by maintaining a *multi-task* GP over $f$, which directly induces correlations between tasks (Bonilla et al., 2008). In the context of BO, Swersky et al. (2013) empirically demonstrate the utility of this model in a number of applications, and Astudillo and Frazier (2019) provide an asymptotic convergence analysis under a special setting of composite objective functions and noise-free evaluations. However, a formal finite time regret analysis showing the effectiveness of multi-task GPs over independent GPs in the context of noisy MOO has not been rigorously pursued. Against this backdrop, we make the following contributions:

- We develop two novel BO algorithms – multi-task kernelized bandits (MT-KB) and multi-task budgeted kernelized bandits (MT-BKB) – that are based on scalarization technique, and can leverage similarities between tasks to optimize them more efficiently.
- Our algorithms use vector-valued kernel ridge regression as a building block and follow the general template of the upper-confidence-bound class of algorithms. In fact, MT-BKB is the first algorithm that employs the *Nyström approximation* technique in the context of *multi-task kernels*.
- Under the assumption that the objective function has smoothness compatible with a joint kernel on its domain and components, we derive regret guarantees for our algorithms that *explicitly capture the inter-task structure*. These are the first worst-case (frequentist) regret bounds for multi-objective BO, and are proved by deriving a novel concentration inequality for the estimate of the vector-valued objective function, which might be of independent interest.
- Finally, our algorithms are simple to implement when the kernel decouples between tasks and domain, and we report numerical results on synthetic as well as real-world based datasets, for which the algorithms are seen to perform favourably.

**Related Work** Popular *multi-objective* BO strategies include Predictive Entropy Search (Hernández-Lobato et al., 2016), max-value entropy search (Belakaria et al., 2019), Pareto active learning (Zuluaga et al., 2013), expected hypervolume improvement (Emmerich and Klinkenberg, 2008), sequential uncertainty reduction (Picheny, 2015) and scalarization based approaches (Knowles, 2006; Zhang and Li, 2007; Paria et al., 2020). Swersky et al. (2013) develop *multi-task* BO strategies with applications in the settings where one cares about learning an expensive primary task based on observations from a cheaper secondary task or transferring the learned knowledge from an already computed task to a new task, hence their framework does not always need simultaneous observations from all tasks. On the other hand, similar to the multi-objective BO strategies mentioned above, we care for optimizing all the tasks together and hence require observations from all of them to ensure a (global) no-regret performance. These strategies, however, model each task independently, and hence, fail to capture any structure present between the tasks. We model all the tasks together using multi-task kernels, and hence describe our framework as "multi-task" optimization.

In the field of geostatistics (Wackernagel, 2013), and more recently in supervised learning (Liu et al., 2018), multi-task GPs and associated kernels have gained a lot of traction. Also, a lot of work has been done in the context of vector-valued or "multi-task" learning with kernel methods (Micchelli and Pontil, 2005; Baldassarre et al., 2012; Grünewälder et al., 2012), and this paper complements the literature by considering an online learning setting. A simple version of multi-objective black box optimization – in the form of online learning in finite multi-armed bandits (MABs) – has been considered in (Drugan and Nowe, 2013; Drugan and Nowé, 2014). This paper, in effect, generalize these works to the more challenging setting of infinite-armed bandits, which has been studied extensively in the single task setting (Srinivas et al., 2010; Chowdhury and Gopalan, 2017; Scarlett et al., 2017).

## 2 PROBLEM STATEMENT

We consider the problem of optimizing a vector-valued function $f(x) = [f_1(x), \ldots, f_n(x)]^\top \in \mathbb{R}^n$ over a compact domain $\mathcal{X} \subset \mathbb{R}^d$ as follows. At each round $t$, a learner queries $f$ at a single point $x_t \in \mathcal{X}$, and observes a noisy output $y_t = f(x_t) + \varepsilon_t$, where $\varepsilon_t \in \mathbb{R}^n$ is a zero-mean $\sigma$-*sub-*

*Gaussian* random vector conditioned on $\mathcal{F}_{t-1}$, the $\sigma$-algebra generated by the random variables $\{x_s, \varepsilon_s\}_{s=1}^{t-1}$ and $x_t$. By this we mean that there exists a $\sigma \geqslant 0$, such that

$$\forall \alpha \in \mathbb{R}^n, \forall t \geqslant 1,\ \mathbb{E}\left[\exp(\alpha^\top \varepsilon_t) \,\middle|\, \mathcal{F}_{t-1}\right] \leqslant \exp\left(\frac{\sigma^2 \|\alpha\|_2^2}{2}\right).$$

The query point $x_t$ at round $t$ is chosen causally depending upon the history $\{(x_s, y_s)\}_{s=1}^{t-1}$ of query and output sequences available up to round $t-1$. Furthermore, it depends on the end user's preference regrarding the trade-offs between different objectives. These preferences are commonly expressed using a *scalarization function*, also known as a utility function, $s_\lambda : \mathbb{R}^n \to \mathbb{R}$, where $\lambda \in \Lambda \subset \mathbb{R}^n$ is a weight vector parameterizing $s_\lambda$ (Roijers et al., 2013). We assume that the scalarization function is $L$-Lipschitz in the $\ell_2$-norm, i.e., $|s_\lambda(u) - s_\lambda(v)| \leqslant L \|u - v\|_2$ for all $\lambda \in \Lambda$ and $u, v \in \mathbb{R}^n$. The Lipschitz constant can be explicitly calculated for commonly used scalarization functions, e.g., the linear scalarization $s_\lambda(y) = \sum_{i=1}^n \lambda_i y_i$ and the Chebyshev scalarization $s_\lambda(y) = \min_{i \leqslant n} \lambda_i (y_i - z_i)$, where $\lambda$ lies in the set $\Lambda = \{\lambda \in \mathbb{R}^n : \lambda_i > 0\ \forall i \leqslant n, \|\lambda\|_1 = 1\}$ and $z \in \mathbb{R}^n$ is a reference point (Nakayama et al., 2009).

**Regularity Assumptions** If the user preference or equivalently, the weight vector $\lambda$ is known beforehand, it is possible to apriori scalarize the objective vectors $f(x)$ and apply standard single-objective BO algorithms. However, often we do not know $\lambda$ in advance and hence, in such cases we need a model that expresses the multiple objectives explicitly. As discussed before, existing works those model each scalar objective $f_i$ independently fail to capture the structure present between different objectives. In contrast, we propose to capture the inter-task structure by modelling the vector-valued function $f$ directly.

**Multi-task Kernel and Its RKHS** We call a mapping $\Gamma : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{n \times n}$, a *multi-task kernel* on $\mathcal{X}$ if $\Gamma(x, x')^\top = \Gamma(x', x)$ for any $x, x' \in \mathcal{X}$, and $\sum_{i,j=1}^m y_i^\top \Gamma(x_i, x_j) y_j \geqslant 0$ for $x_i \in \mathcal{X}$ and $y_i \in \mathbb{R}^n$ for all $i \in [m]$, $m \in \mathbb{N}$.[1] Given a continuous (relative to the induced matrix norm) multi-task kernel $\Gamma$ on $\mathcal{X}$, there exists a unique (modulo an isometry) vector-valued reproducing kernel Hilbert space (RKHS) of vector-valued continuous functions $g : \mathcal{X} \to \mathbb{R}^n$, with $\Gamma$ as its reproducing kernel (Carmeli et al., 2010). We denote this RKHS as $\mathcal{H}_\Gamma(\mathcal{X})$, with the corresponding inner product $\langle \cdot, \cdot \rangle_\Gamma$. Then, for every $x \in \mathcal{X}$, there exists a bounded linear operator $\Gamma_x : \mathbb{R}^n \to \mathcal{H}_\Gamma(\mathcal{X})$ such that $\Gamma(x, x') = \Gamma_x^\top \Gamma_{x'}$ for all $x' \in \mathcal{X}$ and $g(x) = \Gamma_x^\top g$ for all $g \in \mathcal{H}_\Gamma(\mathcal{X})$. Here, $\Gamma_x^\top$ denotes the adjoint of $\Gamma_x$ (with a slight abuse of notation), and it is the unique operator satisfying $\langle \Gamma_x^\top g, y \rangle_2 = \langle g, \Gamma_x y \rangle_\Gamma$ for all $g \in \mathcal{H}_\Gamma(\mathcal{X})$ and $y \in \mathbb{R}^n$. We assume that the objective function $f$ is an element of the RKHS $\mathcal{H}_\Gamma(\mathcal{X})$ and its norm associated to $\mathcal{H}_\Gamma(\mathcal{X})$ is bounded, i.e., there exists a $b < \infty$ such that $\|f\|_\Gamma \leqslant b$. This is a measure of smoothness of $f$, since, by the reproducing property, $\|f(x) - f(x')\|_2 \leqslant \|f\|_\Gamma \|\Gamma_x - \Gamma_{x'}\|$, where $\|\Gamma_x\| := \sup_{\|y\|_2 \leqslant 1} \|\Gamma_x y\|_\Gamma$ denotes the operator norm. Further, we assume that there exists a $\kappa < \infty$ such that $\|\Gamma(x, x)\| \leqslant \kappa$ for all $x \in \mathcal{X}$. Note that in the single-task setting ($n = 1$), the kernel $\Gamma$ is scalar-valued and the RKHS $\mathcal{H}_\Gamma(\mathcal{X})$ consists of real-valued functions. In this case, the bounded norm assumption holds for stationary kernels, e.g., the *squared exponential* (SE) kernel and the *Matérn* kernel (Srinivas et al., 2010).

[1] In its more general form, this definition can be lifted from $\mathbb{R}^n$ to any arbitrary Hilbert space $\mathcal{H}$ (Caponnetto et al., 2008).

**Examples of Multi-task Kernels** It is possible to construct multi-task (MT) kernels using scalar kernels $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$. Evgeniou et al. (2005) consider the kernel $\Gamma(x, x') = k(x, x')\,(\omega I_n + (1 - \omega) 1_n / n)$, where $I_n$ is the $n \times n$ identity matrix, $1_n$ is the $n \times n$ all-one matrix and $\omega \in [0, 1]$ is a parameter that governs the similarity level between components of $f$. The choice $\omega = 1$ corresponds to assuming that all tasks are unrelated and possible similarity among them is not exploited. Conversely, $\omega = 0$ is equivalent to assuming that all tasks are identical and can be explained by the same function. Swersky et al. (2013) consider a more general class of kernels known as the *intrinsic coregionalization model* (ICM), which includes the aforementioned kernel as a special case. The kernels are of the form $\Gamma(x, x') = k(x, x')B$, where $B$ is an $n \times n$ p.s.d. matrix that encodes the inter-task structure and can model both positive or negative correlation between different tasks. This class of kernels is called *separable* since it allows to decouple the contribution of input and output in the covariance structure (Alvarez et al., 2011). We consider stationary scalar kernels $k$ with unit variances – to avoid redundancy in the parameterization – since the variances can be captured fully by $B$ (Bonilla et al., 2008). The main advantage of ICM is that one can use the eigen-system of $B$ to define a new coordinate system where $\Gamma$ becomes block diagonal, reducing the computational burden to a great extent. The diagonal MT kernel $\Gamma(x, x') = \operatorname{diag}(k_1(x, x'), \ldots, k_n(x, x'))$ has the same advantage, but corresponds to treating each task independently using different scalar kernels $k_j$. However, in general, a MT kernel will not be diagonal, and moreover cannot be reduced to a diagonal one by linearly transforming the output space. For example, it is impossible to reduce the kernel $\Gamma(x, x') = \sum_{j=1}^M k_j(x, x') B_j$, $M \neq 1$, to a diagonal one, unless all the $n \times n$ matrices $B_j$ are simultaneously diagonalizable (Caponnetto et al., 2008).

**Performance Metric** The difficulty of specifying the exact scalarization is apparent when the end user is not a single person but a certain population whose members have different preferences regarding different objectives. In such settings, it is desirable to design algorithms that can perform well on average across the entire population of users, i.e., for a family of scalarization functions. To this end, we assume a (known) prior distribution $P_\lambda$ with support on $\Lambda$,

the set of preferences (weights) of all users. This intuitively translates to a prior over the set of scalarizations $(s_\lambda)_{\lambda\in\Lambda}$. (Note that the Dirac-delta distribution[2] yields a deterministic scalarization.) We consider the *optimization* goal and define the expected *utility* of a decision $x$ for a scalarization function $s_\lambda$ and probability distribution $P_\lambda$ as $\mathbb{E}\left[s_\lambda(f(x))\right]$. The learner's performance over a time budget $T$ is measured by the *cumulative regret*, defined as

$$R_C(T) = \sum_{t=1}^{T} \mathbb{E}\left[s_\lambda\left(f(x^\star)\right) - s_\lambda\left(f\left(x_t\right)\right)\right],$$

where $x^\star = \operatorname{argmax}_{x\in\mathcal{X}} \mathbb{E}\left[s_\lambda\left(f(x^\star)\right)\right]$ is the decision that fetches the maximum expected utility. The regret measures the amount of expected utility the learner gives up by not knowing the function $f$ in advance and and taking the optimal decision from the start. We will be concerned with algorithms that attain sublinear regret $R_C(T) = o(T)$ in the number of rounds they face, since, for instance, an algorithm that does not adapt its decisions depending on past experience can easily be seen to achieve linear ($\Omega(T)$) regret.

## 3 OUR APPROACH

We follow the general template of upper confidence bound (UCB) class of BO algorithms (Srinivas et al., 2010) suitably adapted to the multi-task setting. At each round $t$, we compute a multi-task acquisition function $u_t : \mathcal{X} \to \mathbb{R}$ to act as an UCB for the expected (w.r.t. the prior $P_\lambda$) scalarized objective $\mathbb{E}\left[s_\lambda(f(x))\right]$. Whenever $u_t(x)$ is a valid UCB, i.e., $\mathbb{E}\left[s_\lambda(f(x))\right] \leqslant u_t(x)$, and it converges to $s_\lambda\left(f(x)\right)$ "sufficiently" fast, then selecting candidates that are optimal with respect to $u_t$ leads to low cumulative regret, i.e., the expected utility $\mathbb{E}\left[s_\lambda(f(x_t))\right]$ at $x_t \in \operatorname{argmax}_{x\in\mathcal{X}} u_t(x)$ tends to the optimal expected utility $\mathbb{E}\left[s_\lambda(f(x^\star))\right]$ as $t$ increases. It now remains to design a principled multi-task acquisition function $u_t$ based on the scalarization $s_{\lambda_t}$, and in what follows, we shall describe two algorithms for that.

### 3.1 Algorithm 1: Multi-task Kernelized Bandits (MT-KB)

Given the data $\{(x_i, y_i)\}_{i=1}^t \subset \mathcal{X} \times \mathbb{R}^n$, we first aim to find an estimate of $f$ by solving a vector-valued regression problem:

$$\min_{f\in\mathcal{H}_\Gamma(\mathcal{X})} \sum_{i=1}^{t} \left\|y_i - f(x_i)\right\|_2^2 + \eta \left\|f\right\|_\Gamma^2 ,$$

where $\eta > 0$ is a regularizing parameter. Micchelli and Pontil (2005) show that the solution of this minimization problem can be written as $\mu_t = \sum_{i=1}^t \Gamma_{x_i}\alpha_i$. Here, $\{\alpha_i\}_{i=1}^t \subseteq \mathbb{R}^n$ is the unique solution of the linear system of equations $\sum_{i=1}^t \left(\Gamma(x_j, x_i) + \eta\delta_{j,i}\right)\alpha_i = y_j$, $1 \leqslant j \leqslant t$, where $\delta_{j,i}$ denotes the Kronecker-delta function. Now, by the reproducing property, we have

$$\mu_t(x) = \Gamma_x^\top \mu_t = G_t(x)^\top (G_t + \eta I_{nt})^{-1} Y_t, \qquad (1)$$

where the kernel matrix $G_t = \left[\Gamma(x_i, x_j)\right]_{i,j=1}^t$ is a $t \times t$ block matrix with each block being an $n \times n$ matrix (so that $G_t$ is an $nt \times nt$ matrix), $Y_t = \left[y_1^\top, \ldots, y_t^\top\right]^\top$ is an $nt \times 1$ vector with the outputs concatenated, and $G_t(x) = \left[\Gamma(x, x_1)^\top, \ldots, \Gamma(x, x_t)^\top\right]^\top$ is an $nt \times n$ matrix. Notice that $G_t(x)$ can be interpreted as an embedding of a point $x$ supported over the points $x_1, \ldots, x_t$ observed so far. Now, if an arm $x$ is sufficiently unexplored, the estimate $\mu_t(x)$ will, in general, have high variance. One natural way of specifying the uncertainty around $\mu_t(x)$ is the following multi-task kernel:

$$\Gamma_t(x, x') = \Gamma(x, x') - G_t(x)^\top (G_t + \eta I_{nt})^{-1} G_t(x') , \qquad (2)$$

To see this, we draw a connection to multi-task Gaussian processes (MT-GPs) (Liu et al., 2018). Let $f \sim \mathcal{GP}(0, \Gamma)$ be a sample from a zero-mean MT-GP with covariance function $\Gamma$ (i.e., $\mathbb{E}\left[f_i(x)\right] = 0$ and $\mathbb{E}\left[f_i(x) f_j(x')\right] = \Gamma(x, x')_{ij}$ for all $i, j \leqslant n$ and $x, x' \in \mathcal{X}$), and assume that the observation noise vectors $\{\varepsilon_t\}_{t\geqslant 1}$ are independent and $\mathcal{N}(0, \eta I_n)$ distributed. Then the posterior distribution of $f$ conditioned on the data $\{(x_i, y_i)\}_{i=1}^t$ is also a MT-GP with mean $\mu_t$ and covariance $\Gamma_t$, yielding a natural uncertainty model. Now, inspired by the optimism-in-face-of-uncertainty principle, we compute the acquisition function for the next round as

$$u_{t+1}(x) = \mathbb{E}\left[s_\lambda\left(\mu_t(x)\right)\right] + L \cdot \beta_t \left\|\Gamma_t(x, x)\right\|^{1/2}, \qquad (3)$$

where $L$ is the Lipschitz constant of the scalarization function. As a result, selecting the arm $x_{t+1}$ with the highest $u_{t+1}$ inherently trades off exploitation, i.e., picking points with high expected utility $\mathbb{E}\left[s_\lambda\left(\mu_t(x)\right)\right]$, with exploration, i.e., picking points with high uncertainty $\left\|\Gamma_t(x, x)\right\|^{1/2}$. The parameter $\beta_t$ balances between these two objectives, and needs be tuned properly to guarantee low regret. (Due to space constraint, we defer the pseudo-code of MT-KB to Appendix A.)

**Computational Complexity of MT-KB** Maximizing the acquisition function $u_t(x)$ over $\mathcal{X}$ is in general NP-hard even for a single task, since it is a highly non-convex function. To simplify the exposition, in what follows, we will assume that an efficient oracle to optimize $u_t(x)$ is provided to us, and the per step cost comes only from computing $u_t(x)$.[3] Now, the cost of computing $u_t(x)$ is dominated by the cost of inversion of the $nt \times nt$ kernel matrix, and thus in principle scales as $O(n^3t^3)$.[4] We note that the cubic dependency with time $t$ is present even in the single-task ($n = 1$) setting (Shahriari et al., 2015) and in this case, in fact, MT-KB reduces to the well-known GP-UCB algorithm

[2] A Dirac-delta is a probability distribution that puts mass 1 on exactly one point in the probability space.

[3] The BO literature offers many techniques to approximately maximize the acquisition function by grid search or Branch and Bound methods (Brochu et al., 2010). (The expectation $\mathbb{E}[s_\lambda(\cdot)]$ can be approximated using classical sampling techniques.)

[4] This can be reduced to $O(n^3t^2)$ using Schur's complement with an additional storage cost of $O(n^2t^2)$.

(Srinivas et al., 2010; Chowdhury and Gopalan, 2017).

**Remark 1** *The diagonal multi-task kernel* $\Gamma(x,x') = \mathrm{diag}\,(k_1(x,x'),\ldots,k_n(x,x'))$ *corresponds to treating each task independently and the problem reduces to inverting* $n$ *kernel matrices yielding a per-step cost of* $O(nt^3)$ *for MT-KB. This is similar to the prior works (Zuluaga et al., 2013; Hernández-Lobato et al., 2016; Belakaria et al., 2019; Paria et al., 2020) which models each task* $f_i$ *as independent samples from scalar Gaussian processes* $\mathcal{GP}(0,k_i)$.

One common approach to improve computational scalability in kernel methods is the Nyström approximation (Drineas and Mahoney, 2005), which restricts the embeddings $G_t(x)$ and the kernel matrix $G_t$ to be supported on a subset (dictionary) $\mathcal{D}_t$ of selected points. However, this can lead to sub-optimal choices and large regret if $\mathcal{D}_t$ is not sufficiently accurate. This brings about a trade-off between larger and more accurate dictionaries, or smaller and more efficient ones. The BKB algorithm solves this for single-task BO (Calandriello et al., 2019). We now generalize BKB for multiple tasks to improve over the $O(n^3t^3)$ cost of MT-KB.

### 3.2 Algorithm 2: Multi-task Budgeted Kernelized Bandits (MT-BKB)

The central idea behind this algorithm is to evaluate an approximate acquisition function $\tilde{u}_t(x)$, which remains a valid UCB over the expected scalarized objective $\mathbb{E}\left[s_\lambda\left(f(x)\right)\right]$ and at the same time is sufficiently close to $u_t(x)$ to ensure low regret. Given the data $\{(x_i,y_i)\}_{i=1}^t$, we start with an empty set (or, as dubbed in Calandriello et al. (2019), dictionary) $\mathcal{D}_t=\emptyset$ and iterate over the set $\{x_1,\ldots,x_t\}$ to update $\mathcal{D}_t$ as follows. For each candidate $x_i$, we compute an inclusion probability $p_{t,i}$, and add $x_i$ to $\mathcal{D}_t$ with probability $p_{t,i}$. The inclusion probabilities $p_{t,i}$ need to be set suitably so that the dictionary is small enough without compromising on its accuracy. Once the sampling is over, let $\mathcal{D}_t$ be given by the set $\{x_{i_1},\ldots,x_{i_{m_t}}\}$, where $m_t$ is the size of $\mathcal{D}_t$ and $i_j\leqslant t$ for each $j\leqslant m_t$. Given the dictionary $\mathcal{D}_t$, let $\tilde{G}_t(x)=\left[\Gamma(x_{i_1},x)^\top/\sqrt{p_{t,i_1}},\ldots,\Gamma(x_{i_{m_t}},x)^\top/\sqrt{p_{t,i_{m_t}}}\right]^\top$ be the $nm_t\times n$ embedding of $x$ supported over all points in $\mathcal{D}_t$ and $\tilde{G}_t=\left[\Gamma(x_{i_u},x_{i_v})/\sqrt{p_{t,i_u}p_{t,i_v}}\right]_{u,v=1}^{m_t}$ be the corresponding $nm_t\times nm_t$ kernel matrix, properly reweighted by the inclusion probabilities. Then we compute the Nyström embeddings as $\tilde{\Phi}_t(x)=\left(\tilde{G}_t^{1/2}\right)^+\tilde{G}_t(x)$, where $(\cdot)^+$ denotes the pseudo-inverse. We now use these embeddings to approximate $\mu_t$ and $\Gamma_t$ as

$$\tilde{\mu}_t(x)=\tilde{\Phi}_t(x)^\top(\tilde{V}_t+\eta I_{nm_t})^{-1}\sum_{s=1}^t\tilde{\Phi}_t(x_s)y_s\ ,$$

$$\tilde{\Gamma}_t(x,x')=\Gamma(x,x')-\tilde{\Phi}_t(x)^\top\tilde{\Phi}_t(x') +\eta\cdot\tilde{\Phi}_t(x)^\top(\tilde{V}_t+\eta I_{nm_t})^{-1}\tilde{\Phi}_t(x')\ ,$$

where $\tilde{V}_t=\sum_{s=1}^t\tilde{\Phi}_t(x_s)\tilde{\Phi}_t(x_s)^\top$ is an $nm_t\times nm_t$ matrix. Finally, similar to (3), we compute the acquisition function for the next round as $\tilde{u}_{t+1}(x)=\mathbb{E}\left[s_\lambda\left(\tilde{\mu}_t(x)\right)\right]+L\cdot\tilde{\beta}_t\|\tilde{\Gamma}_t(x,x)\|^{1/2}$, with $\tilde{\beta}_t$ governing the exploration-exploitation tradeoff. The inclusion probabilities for the next round are computed as $p_{t+1,i}=\min\left\{q\|\tilde{\Gamma}_t(x_i,x_i)\|,1\right\}$, where $q\geqslant 1$ is a parameter trading-off the size of the dictionary and accuracy of the approximation. We note here that constructing $\mathcal{D}_t$ based on approximate posterior variance sampling is well-studied for scalar kernels Alaoui and Mahoney (2015), and in this work, we introduce it for the first time for MT kernels. (Pseudo-code of MT-BKB is deferred to Appendix A.)

**Computational Complexity of MT-BKB** Computing the dictionary involves a linear search over all selected points while the inclusion probabilities are computed already at the previous round, and thus requires $O(t)$ time per step. The Nyström embeddings $\tilde{\Phi}_t(x)$ can be computed in $O(n^3m_t^3)$ time, since an inversion of the matrix $\tilde{G}_t$ is required. By using these embeddings, $\tilde{V}_t$ can now be computed and inverted in $O(n^2m_t^2t)$ and $O(n^3m_t^3)$ time, respectively. Since, in general, $m_t\leqslant t$, the total per step cost of computing the acquisition function $\tilde{u}_t(x)$ is now $O(n^3m_t^2t)$ as opposed to the $O(n^3t^3)$ cost of MT-KB. The computational advantage of MT-BKB is clearly visible when the dictionary size $m_t$ is near constant at every step, i.e., when $m_t=\tilde{O}(1)$, where $\tilde{O}(\cdot)$ hides constant and $\log$ factors. We shall see in Section 4.2 that this holds, for example, for the intrinsic coregionalization model (ICM) with the squared exponential kernel in its scalar part.

### 3.3 Improved Computational Complexity of MT-KB and MT-BKB for ICM Kernels

The computational cost of our algorithms can be greatly reduced for ICM (separable) kernels $\Gamma(x,x')=k(x,x')B$. Let $\{\xi_i\}_{i=1}^n$ be the eigenvalues of $B$ with corresponding orthonormal eigenvectors $\{u_i\}_{i=1}^n$. We then have the kernel matrix $G_t=\sum_{i=1}^n\xi_iK_t\otimes u_iu_i^\top$ and the output vector $Y_t=\sum_{i=1}^nY_t^i\otimes u_i$, where $\otimes$ denotes the Kronecker product, $K_t=[k(x_i,x_j)]_{i,j=1}^t$ is the kernel matrix of the scalar kernel $k$ and $Y_t^i=[y_1^\top u_i,\ldots,y_t^\top u_i]^\top$. Plugging these into (1) and (2), and using properties of Kronecker product, we now obtain $\mu_t(x)=\sum_{i=1}^n\xi_ik_t(x)^\top(\xi_iK_t+\eta I_t)^{-1}Y_t^iu_i$ and $\|\Gamma_t(x,x)\|=\max_{i\leqslant n}\xi_i\left(k(x,x)-\xi_ik_t(x)^\top(\xi_iK_t+\eta I_t)^{-1}k_t(x)\right)$, where $k_t(x)=[k(x_1,x),\ldots,k(x_t,x)]^\top$. We see that the eigen-decomposition of $B$ needs to be computed only once at the beginning and then, in the new coordinate system, we essentially have to solve $n$ independent problems. Specifically, at round $t$, we need to project the vector-valued output $y_t$ to all coordinates and compute $n$ matrix-vector multiplications of size $t$. However, since the kernel matrix $K_t$ is rescaled by the eigenvalues $\xi_i$, we have to perform only one $t\times t$ inversion. Hence, the per-step time complexity of MT-KB is now $O\left(n^2+(n+t)t^2\right)$ as opposed to $O(n^3t^3)$ for general MT kernels. Similarly, the

per-step cost of MT-BKB can be substantially improved to $O\left(n^2+(n+m_t)m_t t\right)$ from the $O(n^3m_t^2t)$ cost in general. Therefore, the kernels of this form allow for a near-linear (in time $t$) per-step cost of MT-BKB at the price of the eigen-decomposition of $B$. This is substantially better than the $O(nt^3)$ per step-cost of existing multi-objective BO approaches (Hernández-Lobato et al., 2016; Paria et al., 2020). (We defer the details to Appendix A.)

**Practical Considerations** Similar to existing works with scalar kernels, our algorithms too need to know the multi-task kernel in advance. It is, however, common practice to randomly sample a small fraction of the design space, and optimize the kernel parameters prior to running BO algorithms (Zuluaga et al., 2013). We also require that the prior distribution $P_\lambda$ to be known. Our algorithms are fully amenable to changing the prior interactively, say by incorporating interactive user feedback similar to the line of Roijers et al. (2017). In fact, our framework allows us to perform a joint posterior inference on the GP model and the weight distribution. We, however, note that this is not the goal of this work and we focus solely on developing theory for multi-task BO.

# 4 THEORETICAL RESULTS

## 4.1 A Concentration Inequality for Vector-valued RKHS function

We now present the first theoretical result of this work, a concentration inequality for the estimate of the unknown multi-objective function $f$, which is then used to prove the regret bounds for our algorithms. Complete proofs of all results presented in this section are deferred to the appendix.

**Theorem 1 (Multi-task concentration)** *Let $f \in \mathcal{H}_\Gamma(\mathcal{X})$ and the noise vectors $\{\varepsilon_t\}_{t\geqslant 1}$ be $\sigma$-sub-Gaussian. Then, for any $\eta>0$ and $\delta\in(0,1]$, with probability at least $1-\delta$, the following holds uniformly over all $x\in\mathcal{X}$ and $t\geqslant 1$ :*

$$\|f(x)-\mu_t(x)\|_2 \leqslant \alpha_t \left\|\Gamma_t(x,x)\right\|^{1/2}, \quad \textit{where}$$

$$\alpha_t = \|f\|_\Gamma + \tfrac{\sigma}{\sqrt{\eta}}\sqrt{2\log(1/\delta)+\log\det(I_{nt}+\eta^{-1}G_t)}.$$

The significance of this bound can be better understood by studying the log-determinant term, and for this, we again draw a connection to MT-GPs. If $f\sim\mathcal{GP}(0,\Gamma)$ and $\varepsilon_t\sim\mathcal{N}(0,\eta I_n)$ i.i.d., then the *mutual information* between $f$ and the outputs $Y_t$ is exactly equal to $\frac{1}{2}\log\det(I_{nt}+\eta^{-1}G_t))$, and it is a measure for the reduction in the uncertainty or, equivalently, the information gain about $f$. Note that while we use GPs to describe the uncertainty in estimating the unknown function $f$, the bound is *frequentist* and does not need any *Bayesian* assumption about $f$. Similar to the single-task setting Durand et al. (2018), the bound is proved by deriving a new self-normalized concentration inequality for martingales in the $\ell_2$ space.[5] We note here that Astudillo and Frazier (2019) consider the much simpler setting of noise-free outputs and their bound can be re-derived as a special case of Theorem 1.

**Remark 2** *The multi-task kernel $\Gamma$ can be seen as a scalar kernel, $\Gamma(x,x')_{ij}=k\left((x,i),(x',j)\right)$, $i,j\leqslant n$, and $G_t$ as an $nt\times nt$ kernel matrix of $k$ evaluated at points $(x_s,i)$, $s\leqslant t$, $i\leqslant n$. In this case, one can use (Chowdhury and Gopalan, 2017, Theorem 2) to derive concentration bounds for each task $f_i$ separately and combine them together to obtain a result similar to Theorem 1 but with a notable change – $\|\Gamma_t(x,x)\|$ being replaced by $\operatorname{tr}(\Gamma_t(x,x))$. Thus, in general, we prove a tighter concentration inequality which eventually leads to a $O(\sqrt{n})$ factor saving in the final regret bound.*

## 4.2 Regret Bound for MT-KB

Theorem 1 allows for a principled way to tune the confidence radii ($\beta_t$, $\tilde{\beta}_t$) of our algorithms and achieve low regret. We now present the regret bound of MT-KB, which, to the best of our knowledge, is the first frequentist regret guarantee for multi-task BO under a general MT kernel.

**Theorem 2 (Cumulative regret of MT-KB)** *Let $f\in\mathcal{H}_\Gamma(\mathcal{X})$, $\|f\|_\Gamma\leqslant b$ and $\|\Gamma(x,x)\|\leqslant\kappa$ for all $x$. Let $s_\lambda$ be L-Lipschitz and $\{\varepsilon_t\}_{t\geqslant 1}$ be $\sigma$-sub-Gaussian. Then, for any $\eta>0$ and $\delta\in(0,1]$, MT-KB with $\beta_t = b+\frac{\sigma}{\sqrt{\eta}}\sqrt{2\log(1/\delta)+\sum_{s=1}^t\log\det\left(I_n+\eta^{-1}\Gamma_{s-1}(x_s,x_s)\right)}$, enjoys, with probability at least $1-\delta$, the regret bound*

$$R_C^{\textit{MT-KB}}(T)\leqslant 2L\left(b+\frac{\sigma}{\sqrt{\eta}}\sqrt{2\log(1/\delta)+\gamma_{nT}(\Gamma,\eta)}\right)\times\sqrt{(1+\kappa/\eta)T\sum_{t=1}^T\|\Gamma_t(x_t,x_t)\|},$$

*where $\gamma_{nT}(\Gamma,\eta):=\max_{\mathcal{X}_T\subset\mathcal{X}}\frac{1}{2}\log\det\left(I_{nT}+\eta^{-1}G_T\right)$ denotes the maximum information gain.*

Theorem 2, along with the upper bound $\sum_{t=1}^T\|\Gamma_t(x_t,x_t)\|\leqslant 2\eta\gamma_{nT}(\Gamma,\eta)$, yields the more compact regret bound $\tilde{O}\big(b\sqrt{T\gamma_{nT}(\Gamma,\eta)}+\gamma_{nT}(\Gamma,\eta)\sqrt{T}\big)$. We note here that the bound for single-task case (Chowdhury and Gopalan, 2017) can be recovered by setting $n=1$. Furthermore, since the single-task bound is shown to be tight upto a poly-logarithmic factor (Scarlett et al., 2017), our bound, we believe, is also tight in terms of dependence on $T$. Now, we instantiate Theorem 2 for separable MT kernels to point out the novel insights and improvements that our analysis unearths as compared to existing work.

[5] Theorem 1 can even be generalized to the regime of infinite-task learning (Kadri et al., 2016; Brault et al., 2019), where the observations lie in a Hilbert space $\mathcal{H}$, and thus can be of independent interest. The only technical assumption that one will need is that the multi-task kernel $\Gamma(x,x)$ has a finite trace, which trivially holds in the finite-task setting.

**Lemma 1 (Inter-task structure in regret bound)** *Let $B$ be an $n\times n$ p.s.d. matrix and $\Gamma(x,x')=k(x,x')B$. Let $\|\Gamma(x,x)\|\leqslant\kappa$ and $k(x,x)=1$ for all $x\in\mathcal{X}$. Then*

$$\gamma_{nT}(\Gamma,\eta)\leqslant\sum\nolimits_{i\in[n]:\xi_i>0}\gamma_T(k,\eta/\xi_i)\ , \quad \textit{and}$$

$$\sum\nolimits_{t=1}^{T}\|\Gamma_t(x_t,x_t)\|\leqslant 2\eta\max\{\kappa,1\}\gamma_T(k,\eta)\ ,$$

*where $\xi_1,\ldots,\xi_n$ are the eigenvalues of $B$ and $\gamma_T(k,\alpha):=\max_{\mathcal{X}_T\subset\mathcal{X}}\frac{1}{2}\log\det\left(I_T+\alpha^{-1}K_T\right)$, $\alpha>0$, is the maximum information gain associated with the scalar kernel $k$.*

While information gain is a well understood quantity for scalar kernels (Srinivas et al., 2010), Lemma 1 aims to characterize it for multi-task kernels. In particular, we show that information gain (and, therefore, our regret bound) for separable MT kernels can effectively capture the structure between different tasks by means of the spectral properties of the task-similarity matrix $B$. For example, consider the case $B=\omega I_n+(1-\omega)1_n/n$, $\omega\in[0,1]$, which has one eigenvalue equal to 1 and all others equal to $\omega$. In this case, $\gamma_{nT}(\Gamma,\eta)\leqslant\gamma_T(k,\eta)+(n-1)\gamma_T(k,\eta/\omega)$. Now $\gamma_T(k,\eta/\omega)$ is an increasing function in $\omega$, and in fact, $\gamma_T(k,\eta/\omega)=0$ when $\omega=0$. Hence, a low value of $\omega$, i.e., a high amount of similarity between tasks, yields a low cumulative regret and vice-versa. Moreover, for the extreme two cases of $\omega=0$ (all tasks identical) and $\omega=1$ (all tasks unrelated), the regret bounds are $\tilde{O}(\gamma_T(k,\eta)\sqrt{T})$ and $\tilde{O}(\gamma_T(k,\eta)\sqrt{nT})$, respectively. The bounds clearly assert that similar objectives can be learnt much faster together rather than learning them separately. To the best of our knowledge, this intuitive but important observation is not captured by any of the existing theoretical analysis for multi-objective BO (Zuluaga et al., 2013; Belakaria et al., 2019; Paria et al., 2020).

**Remark 3** *Theorem 2 is applicable to any general multi-task kernel, and in the special case of the diagonal kernel $\Gamma(x,x')=\mathrm{diag}\left(k_1(x,x'),\ldots,k_n(x,x')\right)$, yields, along with Lemma 1, a regret bound of $\tilde{O}(\max_i\gamma_T(k_i,\eta)\sqrt{nT})$. This bound, together with the discussion above, suggest that whereas on the one hand MT-KB exploits similarities between tasks efficiently, its performance on the other hand does not suffer when the tasks are unrelated. Another important point to note here is that we analyze the frequentist (worst-case) regret, which is a stronger notion of regret compared to the Bayesian one (defined as the expected cumulative regret under a prior distribution of $f$) as considered in previous works (Belakaria et al., 2019; Paria et al., 2020).*

**Remark 4** *Deshmukh et al. (2017) consider separable multi-task kernels in the finite action contextual bandit setting. While we capture their setting as a special case, their algorithm doesn't work when the action set is continuous.*

### 4.3 Regret and Complexity Bound for MT-BKB

We now present regret and complexity guarantees for MT-BKB, which, to the best of our knowledge, are first of their kinds for multi-task BO under kernel or GP approximation.

**Theorem 3 (Regret bound and complexity of MT-BKB)** *For any $\eta>0$, $\varepsilon\in(0,1)$ and $\delta\in(0,1]$, let $\rho=\frac{1+\varepsilon}{1-\varepsilon}$ and $q=6\rho\log(4T/\delta)/\varepsilon^2$. Then, under the same hypothesis as Theorem 2, if we run MT-BKB with $\tilde{\beta}_t=b\left(1+1/\sqrt{1-\varepsilon}\right)+\frac{\sigma}{\sqrt{\eta}}\sqrt{2\log(2/\delta)+\rho\sum_{s=1}^{t}\log\det\left(I_n+\eta^{-1}\tilde{\Gamma}_{s-1}(x_s,x_s)\right)}$, then, with probability at least $1-\delta$, the following holds:*

$$R_C^{\textit{MT-BKB}}(T)\leqslant 2\rho^{3/2}R_C^{\textit{MT-KB}}(T), \quad \textit{and}$$

$$\forall t\leqslant T,\quad m_t\leqslant 6\rho q(1+\kappa/\eta)\sum\nolimits_{s=1}^{t}\|\Gamma_s(x_s,x_s)\|\ .$$

Theorem 3 shows that MT-BKB can achieve an order-wise similar regret scaling as MT-KB (up to a constant factor), but only at a fraction of the computational cost. To see this, we again consider the kernel $\Gamma(x,x')=k(x,x')B$. In this case, Theorem 3 and Lemma 1 together imply that the dictionary size $m_t$ is $\tilde{O}\left(\gamma_t(k,\eta)\right)$. Now $\gamma_t$ is itself bounded for specific scalar kernels $k$, e.g., it is $O\left((\ln t)^d\right)$ for the squared exponential kernel Srinivas et al. (2010), yielding $m_t$ to be $\tilde{O}(1)$. This leads to a near-linear (in time $t$) per-step cost for MT-BKB compared to the cubic cost for MT-KB. Further, it is worth noting that MT-BKB can adapt to any desired accuracy level $\varepsilon$ of the Nyström approximation. A low value of $\varepsilon$ corresponds to high desired accuracy and MT-BKB adapts to it by inducing more and more points in the dictionary, yielding accurate embeddings and thus, in turn, low regret. Conversely, if one is willing to compromise on the accuracy (given by a high value of $\varepsilon$), then MT-BKB can greatly reduce the size of the dictionary, yielding a low time complexity. The analysis follows in the footsteps of Calandriello et al. (2019), but is carefully generalized to consider multi-task kernels. The regret bound is crucially achieved by showing that $\Gamma_t(x,x)/\rho\preceq\tilde{\Gamma}_t(x,x)\preceq\rho\Gamma_t(x,x)$, i.e., MT-BKB's variance estimates are always almost close to the exact ones ($A\succeq B$ denotes that the matrix $A$–$B$ is p.s.d.). This not only helps us avoid variance starvation which is known to happen with classical sparse GP approximations Wang et al. (2018), but also, allows us to set $\tilde{\beta}_t$ efficiently and in a data-adaptive way.

### 4.4 Pareto Optimality and Random Scalarization

Our results in the previous sections show that both MT-KB and MT-BKB enjoy sublinear upper bound over the cumulative regret. In this section, we briefly discuss and argue that our algorithms can be suitably modified to sample from the whole or a part of the *Pareto front*.

A point $x$ is said to be Pareto dominated by $x'$ if $f(x)\prec f(x')$, i.e., $f_i(x)\leqslant f_i(x')$ for all $i\leqslant n$ and $f_j(x)<f_j(x')$

for some $j \leqslant n$. A point is *Pareto optimal* if it is not Pareto dominated by any other points. The *Pareto front* of $f$ is denoted by $f(\mathcal{X}_f)$, where $\mathcal{X}_f$ is the set of Pareto optimal points and $f(\mathcal{A}) := \{f(x) | x \in \mathcal{A}\}$ for any set $\mathcal{A}$. The goal here is to find a set of $T$ points spanning a particular region of the Pareto front as specified by user preferences encoded in the prior distribution $P_\lambda$. To this end, we consider the *Bayes regret* introduced in Paria et al. (2020) as a performance metric, which is defined as

$$R_B(T) = \mathbb{E}\left[\max_{x \in \mathcal{X}} s_\lambda(f(x)) - \max_{x \in \mathcal{X}_T} s_\lambda(f(x))\right],$$

where $\mathcal{X}_T := \{x_1, \ldots, x_T\}$ is the set of decisions taken by the learner. Paria et al. (2020) argue that in addition to being Lipschitz continuous if the scalarization function is also monotonically increasing in all objectives, i.e., $s_\lambda(f(x)) < s_\lambda(f(x'))$ whenever $f(x) \prec f(x')$, then a low value of Bayes' regret implies that $f(\mathcal{X}_T)$ spans the high probability regions (w.r.t. the prior $P_\lambda$) of the Pareto front $f(\mathcal{X}_f)$. (Monotonicity holds, e.g., for linear and Chebyshev scalarizations.) We refer the interested reader to Paria et al. (2020) for details.

In Appendix E, we show in detail that a simple modification to our algorithms using random scalarizations leads to low Bayes regret. Specifically, at each round $t$, we randomly sample a weight vector $\lambda_t$ from the distribution $P_\lambda$, and replace $\mathbb{E}[s_\lambda(\cdot)]$ in the acquisition function (3) by the random scalarization $s_{\lambda_t}(\cdot)$. We would like to stress that while the approach of Paria et al. (2020) is agnostic to the inter-task structure, our model perfectly captures the similarities present between different tasks. (A numerical comparison is presented in the appendix.)

## 5 EXPERIMENTS

In order to investigate the practical benefits offered by learning with multi-task kernels, we compare MT-KB and MT-BKB with single-task BO algorithms that enjoy regret guarantees under RKHS smoothness assumptions. Specifically, we consider GP-UCB (Chowdhury and Gopalan, 2017) and its Nyström approximate version BKB (Calandriello et al., 2019) as baselines, where each task is learnt independently and inter-task structure is not exploited. We inflate the confidence sets of GP-UCB and BKB properly so that they generalize to multi-objective setting and remain competitive. We call these baselines *independent task kernelized bandits (IT-KB)* and *budgeted kernelized bandits (IT-BKB)*, respectively. Furthermore, whenever the objective function $f$ is not explicitly generated from an RKHS, we compare our algorithms with the MOBO algorithm of (Paria et al., 2020), which model each task with an independent GP. Since MOBO were originally developed to minimize the Bayes regret ($R_B(T)$), we modify it suitably to tackle the cumulative regret metric ($R_C(T)$) and remain competitive.

In all simulations, we set $\eta = 0.1$, $\delta = 0.1$ and $\varepsilon = 0.5$, and approximate $\mathbb{E}[s_\lambda(\cdot)]$ by a sample average. We follow the approach of Paria et al. (2020) to sample from $P_\lambda$. First, we sample $u$ uniformly from $[0, 1]^n$. Then, we set $\lambda = u / \|u\|_1$ for the linear scalarization and $\lambda = \alpha / \|\alpha\|_1$, where $\alpha_i = \|u\|_1 / u_i$, $i \leqslant n$, for the Chebyshev scalarization. We set the lipschitz constant $L = 1$ for both scalarizations. Whenever $f$ is not generated explicitly from an RKHS, we set $b = \max_{x \in \mathcal{X}} \|f(x)\|_2$. We compare the algorithms on the following MOO problems and plot mean and standard deviation (over 10 independent trials) of the time-average cumulative regret $\frac{1}{T} R_C(T)$.

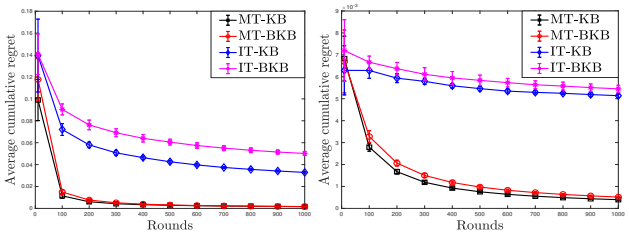


Figure 1: Comparison of average cumulative regret of MT-KB and MT-BKB with IT-KB and IT-BKB under Chebyshev scalarization on functions from RKHS for (a) 2 tasks and (b) 20 tasks.

**RKHS Function** We generate a vector-valued RKHS element as $f(\cdot) = \sum_{i \leqslant 50} \Gamma(\cdot, x_i) c_i$, where the domain $\mathcal{X}$ is an $0.01$-net of the interval $[0, 1]$, each $x_i \in \mathcal{X}$ and each $c_i$ is uniformly sampled from $[-1, 1]^n$. We consider the MT kernel $\Gamma(x, x') = k(x, x') B$ adopting a SE kernel with lengthscale $0.2$ for its scalar part and set $B = A^\top A$, where the elements of the $n \times n$ matrix $A$ is uniformly sampled from $[0, 1]$. The noise vectors are generated i.i.d. $\mathcal{N}(0, \sigma^2 I_n)$ with $\sigma = 0.1$. We compare the algorithms for $n = 2$ and $n = 20$ tasks. We observe that the average cumulative regret decays much rapidly for MT-KB and MT-BKB compared to their respective independent task counterparts IT-KB and IT-BKB which are oblivious to the task similarity matrix $B$ (Fig. 1). This validates our theory that learning with MT kernels is much faster than learning the tasks independently – even more so when number of tasks are higher.

**Perturbed Sine Function** We study a setting similar to Baldassarre et al. (2012), where $\mathcal{X}$ is an $0.01$-net of the interval $[0, 1]$ and there are $n = 4$ tasks. Each task is given by a function $f_i(x) = \sin(2\pi x) + 0.6 f_i^{\text{pert}}(x)$ corrupted by Gaussian noise of variance $0.01$. Each perturbation function $f_i^{\text{pert}}$ is a weighted sum of three Gaussians of width $0.1$ centered at $x_1 = 0.05$, $x_2 = 0.4$ and $x_3 = 0.7$, where task-specific weights are carefully chosen in order to yield tasks that are related by the common function, but also have local differences. We run IT-KB, IT-BKB and MOBO with the SE kernel $k(x, x')$, and MT-KB and MT-BKB with the MT kernel $\Gamma(x, x') = k(x, x') (\omega I_n + (1 - \omega) 1_n / n)$ that imposes a common similarity among all components. We

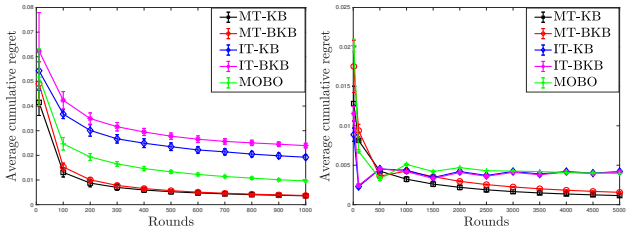


Figure 2: Comparison of average cumulative regret of MT-KB, MT-BKB, IT-KB, IT-BKB and MOBO under Chebyshev scalarization on (a) perturbed sine and (b) shifted Branin-Hoo functions.

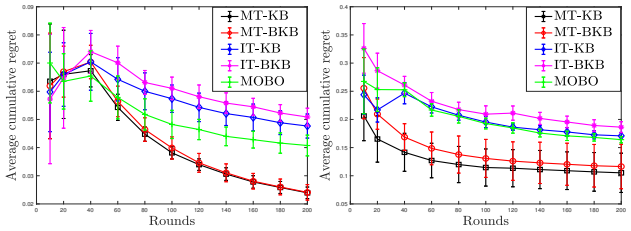


Figure 3: Comparison of average cumulative regret of MT-KB and MT-BKB with IT-KB, IT-BKB and MOBO on sensor data under (a) Chebyshev and (b) linear scalarization.

plot results for $\omega = 0.4$ (Fig. 2).

**Shifted Branin-Hoo** The Branin-Hoo function, defined over a subset of $\mathbb{R}^2$, is a common benchmark for BO (Jones, 2001). We consider 9 shifted Branin-Hoo's as related tasks, where the $i$-th task is a translation of the function by $i\%$ along either axis. We use the same kernels as in the previous experiment and plot results for $\omega = 0.5$ (Fig. 2).

**Sensor measurements** We take temperature, light and humidity measurements from 54 sensors collected in the Intel Berkeley lab (Srinivas et al., 2010). Here we have 3 tasks, one for each variable, and each task $f_i(x)$ is given by the empirical mean of $50\%$ of the readings recorded at the sensor placed at location $x$. We take remaining readings to estimate an ICM (separable) kernel and run our algorithms with this kernel. Specifically, for its scalar part, we fit an SE kernel on sensor locations, and for its matrix part, we estimate inter-task similarities as $B = \frac{1}{m} R^\top K^{-1} R$, where $m$ denotes number of readings, $R$ is an $m \times 3$ matrix of readings for all tasks and $K$ is the $m \times m$ gram matrix of SE kernel. The idea is to de-correlate $R$ with $K^{-1}$ first so that only correlation with respect to $B$ is left. We compute the empirical variance of sensor readings for each task and take the largest of those as the noise variance $\sigma^2$. We see that the regret performance of MT-KB and MT-BKB are much better than IT-KB, IT-BKB and MOBO that do not use the inter-task structure in the form of the matrix $B$ (Fig. 3).

# 6 CONCLUSIONS

To the best of our knowledge, we prove the first rigorous regret bounds for multi-task Bayesian optimization that capture inter-task dependencies. We have demonstrated the shortcoming of modelling each task independently without making use of task similarities, and developed algorithms using multi-task kernels, which perform well in practice. We believe that our regret bounds are tight in terms of dependence on the time horizon. However, whether the dependence on the inter-task structure is optimal or not remains an important open question.

**Acknowledgements** This work has been supported by a Google Ph.D. Fellowship.

## References

Ahmed Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems*, pages 775–783, 2015.

Mauricio A Alvarez, Lorenzo Rosasco, and Neil D Lawrence. Kernels for vector-valued functions: A review. *arXiv preprint arXiv:1106.6251*, 2011.

Francesco Archetti and Antonio Candelieri. *Bayesian optimization and data science*. Springer, 2019.

Raul Astudillo and Peter Frazier. Bayesian optimization of composite functions. In *International Conference on Machine Learning*, pages 354–363, 2019.

Luca Baldassarre, Lorenzo Rosasco, Annalisa Barla, and Alessandro Verri. Multi-output learning via spectral filtering. *Machine learning*, 87(3):259–301, 2012.

Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Max-value entropy search for multi-objective bayesian optimization. In *Advances in Neural Information Processing Systems*, pages 7823–7833, 2019.

Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task gaussian process prediction. In *Advances in neural information processing systems*, pages 153–160, 2008.

Romain Brault, Alex Lambert, Zoltan Szabo, Maxime Sangnier, and Florence d'Alche Buc. Infinite task learning in rkhss. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1294–1302, 2019.

Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

Daniele Calandriello, Luigi Carratino, Alessandro Lazaric, Michal Valko, and Lorenzo Rosasco. Gaussian process optimization with adaptive sketching: Scalable and no regret. *In Conference on Learning Theory*, 2019.

Andrea Caponnetto, Charles A Micchelli, Massimiliano Pontil, and Yiming Ying. Universal multi-task kernels. *Journal of Machine Learning Research*, 9(Jul):1615–1646, 2008.

Claudio Carmeli, Ernesto De Vito, Alessandro Toigo, and Veronica Umanitá. Vector valued reproducing kernel hilbert spaces and universality. *Analysis and Applications*, 8(01):19–61, 2010.

Rich Caruana. Multitask learning. *Machine learning*, 28(1): 41–75, 1997.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 844–853. JMLR. org, 2017.

Aniket Anand Deshmukh, Urun Dogan, and Clay Scott. Multi-task learning for contextual bandits. In *Advances in neural information processing systems*, pages 4848–4856, 2017.

Petros Drineas and Michael W Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *journal of machine learning research*, 6(Dec):2153–2175, 2005.

Madalina M Drugan and Ann Nowe. Designing multi-objective multi-armed bandits algorithms: A study. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2013.

Madalina M Drugan and Ann Nowé. Scalarization based pareto optimal set of arms identification algorithms. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 2690–2697. IEEE, 2014.

Audrey Durand, Odalric-Ambrym Maillard, and Joelle Pineau. Streaming kernel regression with provably adaptive mean, variance, and regularization. *The Journal of Machine Learning Research*, 19(1):650–683, 2018.

Michael Emmerich and Jan-willem Klinkenberg. The computation of the expected improvement in dominated hypervolume of pareto front approximations. *Rapport technique, Leiden University*, 34:7–3, 2008.

Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *Journal of machine learning research*, 6(Apr):615–637, 2005.

Peter I Frazier. Bayesian optimization. In *Recent Advances in Optimization and Modeling of Contemporary Problems*, pages 255–278. INFORMS, 2018.

R. Garnett, M. A. Osborne, and S. J. Roberts. Bayesian optimization for sensor set selection. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '10, pages 209–219, New York, NY, USA, 2010. ACM.

Javier Gonzalez, Joseph Longworth, David C James, and Neil D Lawrence. Bayesian optimization for synthetic gene design. *arXiv preprint arXiv:1505.01627*, 2015.

Robert B Gramacy. *Surrogates: Gaussian process modeling, design, and optimization for the applied sciences*. CRC Press, 2020.

William H Greene. *Econometric analysis*. Pearson Education India, 2003.

Steffen Grünewälder, Guy Lever, Luca Baldassarre, Sam Patterson, Arthur Gretton, and Massimilano Pontil. Conditional mean embeddings as regressors. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1803–1810, 2012.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

Daniel Hernández-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. Predictive entropy search for multi-objective bayesian optimization. In *International Conference on Machine Learning*, pages 1492–1501, 2016.

Donald R Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4):345–383, 2001.

Hachem Kadri, Emmanuel Duflos, Philippe Preux, Stéphane Canu, Alain Rakotomamonjy, and Julien Audiffren. Operator-valued kernels for learning from functional response data. *The Journal of Machine Learning Research*, 17(1):613–666, 2016.

Joshua Knowles. Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

Haitao Liu, Jianfei Cai, and Yew-Soon Ong. Remarks on multi-output gaussian process regression. *Knowledge-Based Systems*, 144:102–121, 2018.

Charles A Micchelli and Massimiliano Pontil. On learning vector-valued functions. *Neural computation*, 17(1):177–204, 2005.

Hirotaka Nakayama, Yeboon Yun, and Min Yoon. *Sequential approximate multiobjective optimization using computational intelligence*. Springer Science & Business Media, 2009.

Biswajit Paria, Kirthevasan Kandasamy, and Barnabás Póczos. A flexible framework for multi-objective bayesian optimization using random scalarizations. In *Uncertainty in Artificial Intelligence*, pages 766–776. PMLR, 2020.

Victor Picheny. Multiobjective optimization using gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, 25(6):1265–1280, 2015.

Ryan Rifkin, Sayan Mukherjee, Pablo Tamayo, Sridhar Ramaswamy, Chen-Hsiang Yeang, Michael Angelo, Michael Reich, Tomaso Poggio, Eric S Lander, Todd R Golub, et al. An analytical method for multiclass molecular cancer classification. *Siam Review*, 45(4):706–723, 2003.

Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

Diederik M Roijers, Luisa M Zintgraf, and Ann Nowé. Interactive thompson sampling for multi-objective multi-armed bandits. In *International Conference on Algorithmic DecisionTheory*, pages 18–34. Springer, 2017.

Jonathan Scarlett, Ilija Bogunovic, and Volkan Cevher. Lower bounds on regret for noisy gaussian process bandit optimization. In *Conference on Learning Theory*, pages 1723–1742, 2017.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 1015–1022. Omnipress, 2010.

Kevin Swersky, Jasper Snoek, and Ryan P Adams. Multi-task bayesian optimization. In *Advances in neural information processing systems*, pages 2004–2012, 2013.

Hans Wackernagel. *Multivariate geostatistics: an introduction with applications*. Springer Science & Business Media, 2013.

Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, pages 745–754, 2018.

Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.

Indre Zliobaite. On the relation between accuracy and fairness in binary classification. *arXiv preprint arXiv:1505.05723*, 2015.

Marcela Zuluaga, Guillaume Sergent, Andreas Krause, and Markus Püschel. Active learning for multi-objective optimization. In *International Conference on Machine Learning*, pages 462–470, 2013.