Beyond Grids: Multi-objective Bayesian Optimization With Adaptive Discretization

Anonymous authors
Paper under double-blind review

Abstract

We consider the problem of optimizing a vector-valued objective function $\boldsymbol{f}$ sampled from a Gaussian Process (GP) whose index set is a well-behaved, compact metric space $(\mathcal{X}, d)$ of designs. We assume that $\boldsymbol{f}$ is not known beforehand and that evaluating $\boldsymbol{f}$ at design x results in a noisy observation of $\boldsymbol{f}(x)$. Since identifying the Pareto optimal designs via exhaustive search is infeasible when the cardinality of $\mathcal{X}$ is large, we propose an algorithm, called Adaptive $\boldsymbol{\epsilon}$-PAL, that exploits the smoothness of the GP-sampled function and the structure of $(\mathcal{X}, d)$ to learn fast. In essence, Adaptive $\boldsymbol{\epsilon}$-PAL employs a tree-based adaptive discretization technique to identify an $\boldsymbol{\epsilon}$-accurate Pareto set of designs in as few evaluations as possible. We provide both information-type and metric dimension-type bounds on the sample complexity of $\boldsymbol{\epsilon}$-accurate Pareto set identification. We also experimentally show that our algorithm outperforms other Pareto set identification methods.

1 Introduction

Many complex scientific problems require the optimization of multi-dimensional (m-variate) performance metrics (objectives) under uncertainty. When developing a new drug, scientists need to identify the optimal therapeutic doses that maximize benefit and tolerability (Scmidt, 1988). When designing new hardware, engineers need to identify the optimal designs that minimize energy consumption and runtime (Almer et al., 2011). In general, there is no design that can simultaneously optimize all objectives, and hence, one seeks to identify the set of Pareto optimal designs. Moreover, design evaluations are costly, and thus, the optimal designs should be identified with as few evaluations as possible. In practice, this is a formidable task for at least two reasons: design evaluations only provide noisy feedback about ground truth objective values, and the set of designs to explore is usually very large (even infinite). Luckily, in practice, one only needs to identify the set of Pareto optimal designs up to a desired level of accuracy. Within this context, a practically achievable goal is to identify an $\boldsymbol{\epsilon}$-accurate Pareto set of designs whose objective values form an $\boldsymbol{\epsilon}$-approximation of the true Pareto front for a given $\boldsymbol{\epsilon} = [\epsilon^1, \dots, \epsilon^m]^\mathsf{T} \in \mathbb{R}^m_+$ (Zuluaga et al., 2016).

There has been a considerable amount of interest in the Pareto set identification problem with a finite design space (Kone et al., 2023; Zuluaga et al., 2013; 2016), where individual designs are compared based on their vector values, according to multi-objective domination criteria. However, in many multi-objective problems of interest, such as the accurate tuning of particle accelerators (Roussel et al., 2021), or robotic systems control (Ariizumi et al., 2014), the design space is not necessarily finite. For these examples, a naive application of methods that are suitable for finite spaces might not yield the desired level of efficiency, due to the lack of rigorous theoretical foundations. Motivated by these observations, in this paper, we consider the Pareto set identification problem assuming a compact design space.

Specifically, we model the identification of an $\boldsymbol{\epsilon}$-accurate Pareto set of designs as an active learning problem. We assume that the designs lie in a well-behaved, compact metric space $(\mathcal{X}, d)$, where the set of designs $\mathcal{X}$ might be very large. The vector-valued objective function $\boldsymbol{f} = [f^1, \dots, f^m]^\mathsf{T}$ defined over $(\mathcal{X}, d)$ is unknown at the beginning of the experiment. The learner is given prior information about $\boldsymbol{f}$, which states that it is a sample from a multi-output GP with known mean and covariance functions. Then, the learner sequentially chooses designs to evaluate, where evaluating $\boldsymbol{f}$ at design x immediately yields a noisy observation of $\boldsymbol{f}(x)$.

The learner uses data from its past evaluations in order to decide which design to evaluate next until it can confidently identify an $\boldsymbol{\epsilon}$-accurate Pareto set of designs.

Main contribution. We propose a new learning algorithm, called Adaptive $\boldsymbol{\epsilon}$-PAL, that solves the Pareto active learning (PAL) problem described above, by performing as few design evaluations as possible. Our algorithm employs a tree-based adaptive discretization strategy to dynamically partition $\mathcal{X}$. It uses the GP posterior on $\boldsymbol{f}$ to decide which regions of designs in the partition of $\mathcal{X}$ to discard or to declare as a member of the $\boldsymbol{\epsilon}$-accurate Pareto set of designs. On termination, Adaptive $\boldsymbol{\epsilon}$-PAL guarantees that the returned set of designs forms an $\boldsymbol{\epsilon}$-accurate Pareto set with a high probability. To the best of our knowledge, Adaptive $\boldsymbol{\epsilon}$-PAL is the first algorithm that employs an adaptive discretization strategy in the context of PAL, which turns out to be very effective when dealing with a large $\mathcal{X}$.

We prove information-type and metric dimension-type upper bounds on the sample complexity of Adaptive $\boldsymbol{\epsilon}$-PAL (Theorem 1). Our information-type bound yields a sample complexity upper bound of $\tilde{O}(g(\epsilon))$ where $\epsilon = \min_j \epsilon^j$, $g(\epsilon) = \min\{\tau \geq 1 : \sqrt{\gamma_\tau/\tau} \leq \epsilon\}$, and γ_τ is the maximum information gain after τ evaluations. To the best of our knowledge, this is the first information type bound for dependent objectives in the context of PAL. In addition, our metric dimension-type bound yields a sample complexity of $\tilde{O}(\epsilon^{-(\frac{\bar{D}}{\alpha}+2)})$ for all $\bar{D} > D_1$, where D_1 represents the metric dimension of $(\mathcal{X}, d)$ and $\alpha \in (0, 1]$ represents the Hölder exponent of the metric induced by the GP on $\mathcal{X}$. As far as we know, this is the first metric dimension-type bound in the context of PAL. Our bounds complement each other: as we show in the appendix, neither of them dominates the other for all possible GP kernels. Specifically, we provide an example (Proposition 1) under which the information-type bound can be very loose compared to the metric dimension-type bound. Besides theory, we also show via simulations on multi-objective functions that Adaptive $\boldsymbol{\epsilon}$-PAL significantly improves over its competitors in terms of accuracy.

Organization. We provide a detailed comparison with related work in Section 2. We explain the properties of the function to be optimized and the structure of the design space in Section 3. This is followed by the description of Adaptive $\boldsymbol{\epsilon}$-PAL in Section 4. We give sample complexity bounds for Adaptive $\boldsymbol{\epsilon}$-PAL in Section 5, discuss the main aspects of computational complexity analysis in Section 6. We devote Section 7 to the computational experiments, followed by conclusions in Section 8. We present the proof of the main theorem separately in Section A, the appendix. At the end of the paper, we include a table for the frequently used notation.

2 Related Work

This section provides a detailed discussion of related work on several lines of research.

2.1 Multi-objective optimization

Learning the Pareto optimal set of designs and the Pareto front has received considerable attention in recent years (Belakaria et al., 2024; Auer et al., 2016; Zuluaga et al., 2013; 2016; Hernández-Lobato et al., 2016; Shah & Ghahramani, 2016; Paria et al., 2020; Belakaria & Deshwal, 2019; Belakaria et al., 2020; Daulton et al., 2020; Alizadeh et al., 2024; Mukherjee et al., 2024).

Auer et al. (2016) consider a finite set of designs and formulate the identification of the Pareto front as a pure exploration multi-armed bandit (MAB) problem in the fixed confidence setting. Under the assumption that the centered outcomes are 1-subGaussian and independent, they provide gap-dependent bounds on its sample complexity, which, in the single objective case, yield the well-known gap-dependent sample-complexity bounds for pure exploration in the multi-armed bandit setting (Mannor & Tsitsiklis, 2004). Other works on Multi-objective MAB include (Xu & Klabjan, 2023; Busa-Fekete et al., 2017; Öner et al., 2018). As opposed to their frequentist approach, our approach is Bayesian, which imposes a Gaussian process prior to the latent function of interest.

Knowles (2006) and Ponweiser et al. (2008) study the multi-objective optimization problem using the paradigm of Efficient Global Optimization (EGO) algorithm (Jones et al., 1998), a GP-based supervised learning approach used to tackle optimization problems with expensive evaluations. Knowles (2006) proposes ParEGO, which is an adaptation of EGO in the m-objective case. The author takes a scalarization approach to the problem, where the used acquisition function is the augmented Tchebycheff function, thus essentially reducing the problem to the single-objective one. Such an acquisition function was also used recently by Lin et al. (2022) for Pareto set learning. They extend previous approaches by identifying an approximate Pareto set of (potentially) infinitely many designs, as opposed to stopping with only a finite number of designs. In contrast, we take a direct Pareto set identification approach, in which we utilize the structure of the partial ordering relation between values of evaluated designs, while simultaneously integrating the GP posterior into the selection conditions of our method.

The second application of EGO, SMS-EGO (Ponweiser et al., 2008), does not reduce the problem to a single-objective one, but instead maximizes the gain in hypervolume from optimistic estimates based on a GP model. Having optimistically estimated the GP-based value vectors of their predicted Pareto set, they compute the hypervolume of the gain of choosing such design points. The hypervolume approach to Pareto learning is further exploited by Shah & Ghahramani (2016) and Daulton et al. (2020). In Shah & Ghahramani (2016), the Pareto hypervolume is defined in terms of an arbitrarily chosen reference point which is known to be suboptimal, and the current Pareto frontier. What makes their method attractive is its efficient implementation and computation, which relies on the approximation of a multi-dimensional integral. The computational complexity of computing the expected hypervolume gain is improved by extending the problem to the parallel constrained evaluation setting in Daulton et al. (2020). However, no one of these methods come with theoretical convergence guarantees, while we provide a comprehensive best-of-both-world type of convergence guarantees under minimal assumptions.

Furthermore, Hernández-Lobato et al. (2016) consider a (possibly infinite) bounded design space $\mathcal{X}$ and assume that the individual objectives are samples from independent GPs. Their algorithm, Predictive Entropy Search for Multi-objective Optimization (PESMO), sequentially queries designs that maximize the acquisition function defined as the expected reduction in the entropy of the posterior distribution over the predicted Pareto set, given the previously sampled data. They provide comprehensive experimental comparisons between PESMO and other multi-objective optimization methods over various objectives and dimensions, in both noisy and noiseless cases. The comparison is done with respect to the relative difference between the hypervolume of the predicted set and the maximum such hypervolume for the given number of evaluations. Although their setup is similar to ours, we take a different approach to the Pareto set identification and provide theoretical guarantees for our method. Recently, Tu et al. (2022) extend the Predictive Entropy Search paradigm to that of Joint Entropy Search (JES), which takes into account the informativeness coming from both the Pareto set designs and their outcome vectors in their acquisition function.

Apart from Pareto front identification, several works consider identifying designs that satisfy certain performance criteria. For instance, Katz-Samuels & Scott (2018) consider the problem of identifying designs whose objective values lie in a given polyhedron in the fixed confidence setting. On the other hand, Locatelli et al. (2016) consider the problem of identifying designs whose objective values are above a given threshold in the fixed budget setting. In addition, Gotovos (2013) consider level set identification when $\boldsymbol{f}$ is a sample from a GP. There also exists a plethora of works developing algorithms for best arm identification in the context of single-objective pure-exploration MAB problems such as the ones by Mannor & Tsitsiklis (2004); Bubeck et al. (2009); Gabillon et al. (2012).

2.2 Adaptive discretization

Adaptive discretization is a technique that is mainly used in regret minimization in MAB problems on metric spaces (Kleinberg et al., 2008; Bubeck et al., 2011), including contextual MAB problems (Shekhar et al., 2018), when dealing with large arm and context sets. It consists of adaptively partitioning the ground set along a tree structure into smaller and smaller regions, until, theoretically, the regions converge to a single point (under uniqueness assumptions in the single-objective case). Different from other upper confidence bound-based methods, here, the usual upper confidence bound of a given point x (in both frequentist and Bayesian approaches) is inflated with a factor times the diameter of the region containing x, so that the uncertainty coming from the variation of the function values inside the region is also captured. A key efficacy of adaptive discretization comes from the fact that it does not blindly sample the space without first exhaustively partitioning it as long as it is certain. This certainty is formalized by comparing the sample

Table 1: *Comparison with related works.* [(1)]Both the algorithm and the performance analysis take into account dependence between the objectives.

Work	**Design space**	**Function**	**Dep.**[(1)]	**Sample complex. bounds**	**Adaptive discret.**
Auer et al. (2016)	Finite	Arbitrary	No	Gap-dependent	Not used
Zuluaga et al. (2013)	Finite	GP sample	No	Inf.-type	Not used
Zuluaga et al. (2016)	Finite	RKHS element	No	Inf.-type	Not used
Hernández-Lobato et al. (2016)	Bounded	GP sample	No	No bound	Not used
Shah & Ghahramani (2016)	Bounded	GP sample	Yes	No bound	Not used
Paria et al. (2020)	Compact	GP sample	No	Bayes Regret	Not used
Belakaria & Deshwal (2019)	General	GP sample	No	Regret norm	Not used
Belakaria et al. (2020)	Continuous	GP sample	No	Regret norm	Not used
Daulton et al. (2020)	Bounded	GP sample	No	Exp. Hypervol.	Not used
Ours	Compact	A GP sample	Yes	Inf. & dim.-type	Used

uncertainty diameter with the region diameter. If the latter exceeds the former, then the algorithm decides to partition the given region into smaller sub-regions. It is known that adaptive discretization can result in a much smaller regret compared to uniform discretization. However, this is significantly different from employing adaptive discretization in the context of PAL. While the regret can be minimized by quickly identifying one design that yields the highest expected reward, PAL requires identifying all designs that can form an $\boldsymbol{\epsilon}$-Pareto front together, while at the same time discarding all designs that are far from being Pareto optimal. Table 1 compares our approach with the related work.

2.3 ϵ-Pareto active learning

The line of work on which our method builds is that of Zuluaga et al. (2013; 2016). Zuluaga et al. (2016) consider a finite design space and assume that each objective is a sample from an independent GP. We briefly describe the rationale of their algorithm, since it will also be used later on. Their algorithm, ϵ-Pareto Active Learning (ϵ-PAL), is a confidence bound-based method. It partitions the design space into three subsets: the set of undecided designs, that of predicted Pareto-optimal designs, and that of predicted suboptimal designs. The procedure is composed of four main phases. In the first phase, the algorithm uses the posterior estimates in order to compute the confidence hyper-rectangles of designs that are still up for selection. In the second phase, each design is associated with an uncertainty region, dependent on all previous confidence hyper-rectangles, and then checked whether it is safe to discard it. The third phase consists of deciding whether there are any designs that can be safely predicted to be Pareto optimal. In the final phase, ϵ-PAL decides whether or not to evaluate the design of maximal uncertainty.

The algorithm we propose utilizes a similar rationale to that of ϵ-PAL. However, such an extension, although seemingly natural, also brings with it several challenges and key differences, which necessitate novel ways of solving the problem. We summarize these differences in the following.

First, ϵ-PAL iterates through all designs, maintaining relevant statistics of them which it uses to decide whether they are reasonable candidates for Pareto optimality, or whether they can be safely discarded. In our case, this would be a futile attempt, since the design space is potentially infinite. Therefore, we use adaptive discretization techniques (Bubeck et al., 2011) in order to tackle the problem. Next, the application of adaptive discretization in the context of PAL introduces additional technical intricacies: instead of working over individual designs, our method maintains statistics over nodes (centers) of design regions with similar values, and, as a result, with similar levels of confidence. Thus, we design novel bonus terms for each node, which simultaneously take into account both the hyper-rectangular uncertainty and the structural relationship of "nearby"[1] nodes.

[1]We assume a tree structure defined over the design space, where parent-child relationships are properly defined between relevant nodes. See Definition 6.

On the algorithmic front, the successful integration of adaptive discretization into the PAL setting implies a new evaluation procedure. We part from the ϵ-PAL evaluation subroutine and introduce a new condition, which captures the uncertainty over a given region and implies that the algorithm will evaluate a design from the region only when it is absolutely necessary, thus optimizing the sample complexity of the overall procedure. On the theoretical front, on top of the information-type sample complexity bounds provided by Zuluaga et al. (2016), we additionally provide dimension-type bounds, thus yielding best-of-both-worlds bounds. We do this motivated by examples in which the information-type bounds might actually be loose (see Proposition 1). To make our theoretical analysis rigorous, we prove several additional results that allow for a comprehensive understanding of our bounds. To the best of our knowledge, we are the first to provide such a full comprehensive analysis for both types of bounds, while simultaneously accounting for adaptive discretization in the context of PAL.

3 Background and formulation

Throughout the paper, let us fix a positive integer $m \geq 2$ and a compact metric space $(\mathcal{X}, d)$. We denote by $\mathbb{R}^m$ the m-dimensional Euclidean space and by $\mathbb{R}^m_+$ the set of all vectors in $\mathbb{R}^m$ with nonnegative components. We write $[m] = \{1, \dots, m\}$. Given a function $\boldsymbol{f}\colon \mathcal{X} \to \mathbb{R}^m$ and a set $\mathcal{S} \subseteq \mathcal{X}$, we denote by $\boldsymbol{f}(\mathcal{S}) = \{\boldsymbol{f}(x)\colon x \in \mathcal{S}\}$ the image of $\mathcal{S}$ under $\boldsymbol{f}$. Given $x \in \mathcal{X}$ and $r \geq 0$, $B(x,r) = \{y \in \mathcal{X}\colon d(x,y) \leq r\}$ denotes the closed ball centered at x with radius r. For a non-empty set $\mathcal{S} \subseteq \mathbb{R}^m$, let $\partial\mathcal{S}$ denote its boundary. If another non-empty set $\mathcal{S}' \subseteq \mathbb{R}^m$ is given, then we define the Minkowski sum and difference of $\mathcal{S}$ and $\mathcal{S}'$ as $\mathcal{S} + \mathcal{S}' = \{\boldsymbol{\mu} + \boldsymbol{\mu}'\colon \boldsymbol{\mu} \in \mathcal{S}, \boldsymbol{\mu}' \in \mathcal{S}'\}$, $\mathcal{S} - \mathcal{S}' = \{\boldsymbol{\mu} - \boldsymbol{\mu}'\colon \boldsymbol{\mu} \in \mathcal{S}, \boldsymbol{\mu}' \in \mathcal{S}'\}$, respectively. For a vector $\boldsymbol{\mu}'' \in \mathbb{R}^m$, we define $\boldsymbol{\mu}'' + \mathcal{S} = \{\boldsymbol{\mu}''\} + \mathcal{S}$.

3.1 Multi-objective optimization

A multi-objective optimization problem is an optimization problem that involves multiple objective functions (Hwang & Masud, 2012). Formally, letting $f^j : \mathcal{X} \to \mathbb{R}$ be a function for every $j \in [m]$, we write

$$\text{maximize } [f^1(x), \dots, f^m(x)]^\mathsf{T} \text{ subject to } \ x \in \mathcal{X},$$

where $m \geq 2$ is the number of objectives and $\mathcal{X}$ is the set of designs. We refer to the vector of all objectives evaluated at design $x \in \mathcal{X}$ as $\boldsymbol{f}(x) = [f^1(x), \dots, f^m(x)]^\mathsf{T}$. The objective space is given as $\boldsymbol{f}(\mathcal{X}) \subseteq \mathbb{R}^m$. In order to define a set of Pareto optimal designs in $\mathcal{X}$, we first describe several order relations on $\mathbb{R}^m$.

Definition 1. *For $\boldsymbol{\mu}, \boldsymbol{\mu}' \in \mathbb{R}^m$, we say that: (1) $\boldsymbol{\mu}$ is weakly dominated by $\boldsymbol{\mu}'$, written as $\boldsymbol{\mu} \preceq \boldsymbol{\mu}'$, if $\mu_j \leq \mu'_j$ for every $j \in [m]$. (2) $\boldsymbol{\mu}$ is dominated by $\boldsymbol{\mu}'$, written as $\boldsymbol{\mu} \prec \boldsymbol{\mu}'$, if $\boldsymbol{\mu} \preceq \boldsymbol{\mu}'$ and there exists $j \in [m]$ with $\mu_j < \mu'_j$. (3) For $\boldsymbol{\epsilon} \in \mathbb{R}^m_+$, $\boldsymbol{\mu}$ is $\boldsymbol{\epsilon}$-dominated by $\boldsymbol{\mu}'$, written as $\boldsymbol{\mu} \preceq_{\boldsymbol{\epsilon}} \boldsymbol{\mu}'$, if $\boldsymbol{\mu} \preceq \boldsymbol{\mu}' + \boldsymbol{\epsilon}$. (4) $\boldsymbol{\mu}$ is incomparable with $\boldsymbol{\mu}'$, written as $\boldsymbol{\mu} \parallel \boldsymbol{\mu}'$, if neither $\boldsymbol{\mu} \prec \boldsymbol{\mu}'$ nor $\boldsymbol{\mu}' \prec \boldsymbol{\mu}$ holds.*

Based on Definition 1, we define the following induced relations on $\mathcal{X}$.

Definition 2. *For designs $x, y \in \mathcal{X}$, we say that: (1) x is weakly dominated by y, written as $x \preceq y$, if $\boldsymbol{f}(x) \preceq \boldsymbol{f}(y)$. (2) x is dominated by y, written as $x \prec y$, if $\boldsymbol{f}(x) \prec \boldsymbol{f}(y)$. (3) For $\boldsymbol{\epsilon} \in \mathbb{R}^m_+$, x is $\boldsymbol{\epsilon}$-dominated by y, written as $x \preceq_{\boldsymbol{\epsilon}} y$, if $\boldsymbol{f}(x) \preceq_{\boldsymbol{\epsilon}} \boldsymbol{f}(y)$. (4) x is incomparable with y, written as $x \parallel y$, if $\boldsymbol{f}(x) \parallel \boldsymbol{f}(y)$.*

Note that, while the relation $\preceq$ on $\mathbb{R}^m$ (Definition 1) is a partial order, the induced relation $\preceq$ on $\mathcal{X}$ (Definition 2) is only a preorder since it does not satisfy antisymmetry in general. If a design $x \in \mathcal{X}$ is not dominated by any other design, then we say that x is *Pareto optimal.* The set of all Pareto optimal designs is called the *Pareto set* and is denoted by $\mathcal{O}(\mathcal{X})$. The *Pareto front* is defined as $\mathcal{Z}(\mathcal{X}) = \partial(\boldsymbol{f}(\mathcal{O}(\mathcal{X})) - \mathbb{R}^m_+)$.

We assume that $\boldsymbol{f}$ is not known beforehand and formalize the goal of identifying $\mathcal{O}(\mathcal{X})$ as a sequential decision-making problem. In particular, we assume that evaluating $\boldsymbol{f}$ at design x results in a noisy observation of $\boldsymbol{f}(x)$. The exact identification of the Pareto set and the Pareto front using a small number of evaluations is, in general, not possible under this setup, especially when the cardinality of $\mathcal{X}$ is infinite or a very large finite number. A realistic goal is to identify the Pareto set and the Pareto front in an approximate sense, given a desired level of accuracy that can be specified as an input $\boldsymbol{\epsilon}$. Therefore, our goal in this paper is to identify an $\boldsymbol{\epsilon}$-accurate Pareto set (see Definition 5) that contains a set of near-Pareto optimal designs by using as few evaluations as possible. Next, we define the $\boldsymbol{\epsilon}$-Pareto front and $\boldsymbol{\epsilon}$-accurate Pareto set associated with $\mathcal{X}$.

Definition 3. *Given $\boldsymbol{\epsilon} \in \mathbb{R}_+^m$, the set $\mathcal{Z}_{\boldsymbol{\epsilon}}(\mathcal{X}) = (\boldsymbol{f}(\mathcal{O}(\mathcal{X})) - \mathbb{R}_+^m) \setminus (\boldsymbol{f}(\mathcal{O}(\mathcal{X})) - 2\boldsymbol{\epsilon} - \mathbb{R}_+^m)$ is called the **$\boldsymbol{\epsilon}$-Pareto front** of $\mathcal{X}$.*

Roughly speaking, the $\boldsymbol{\epsilon}$-Pareto front can be thought of as the slab of points of width $2\boldsymbol{\epsilon}$ in $\mathbb{R}^m$ adjoined to the lower side of the Pareto front.

Definition 4. *Given $\boldsymbol{\epsilon} \in \mathbb{R}_+^m$ and $\mathcal{S} \subseteq \mathbb{R}^m$, a non-empty subset $\mathcal{C}$ of $\mathcal{S}$ is called an **$\boldsymbol{\epsilon}$-covering of $\mathcal{S}$** if for every $\boldsymbol{\mu} \in \mathcal{S}$, there exists $\boldsymbol{\mu}' \in \mathcal{C}$ such that $\boldsymbol{\mu} \preceq_{\boldsymbol{\epsilon}} \boldsymbol{\mu}'$.*

Definition 5. *Given $\boldsymbol{\epsilon} \in \mathbb{R}_+^m$, a subset $\mathcal{O}_{\boldsymbol{\epsilon}}$ of $\mathcal{X}$ is called an **$\boldsymbol{\epsilon}$-accurate Pareto set** if $\boldsymbol{f}(\mathcal{O}_{\boldsymbol{\epsilon}})$ is an $\boldsymbol{\epsilon}$-covering of $\mathcal{Z}_{\boldsymbol{\epsilon}}(\mathcal{X})$.*

Note that the front associated with an $\boldsymbol{\epsilon}$-accurate Pareto set is a subset of the $\boldsymbol{\epsilon}$-Pareto front. As mentioned in Zuluaga et al. (2016), an $\boldsymbol{\epsilon}$-accurate Pareto set is a natural substitute of the Pareto set since any $\boldsymbol{\epsilon}$-accurate Pareto design is guaranteed to be no worse than $2\boldsymbol{\epsilon}$ of any Pareto optimal design.

3.2 Structure and dimensionality of the design space

In order to learn, we need to make use of some concepts that facilitate understanding the structure of the space and 'navigating' it. To that end, we will assume that the design space is *well-behaved.* This will allow us to partition the design space $\mathcal{X}$ along a tree structure, each level h of which is associated with a partition of $\mathcal{X}$ into N^h equal-sized regions $X_{h,i}$, centered at a node $x_{h,i}$, for all $0 \leq i \leq N^h$, where $N \in \mathbb{N}$. The formal definition is as follows.

Definition 6. *(Well-behaved metric space (Bubeck et al., 2011)) The compact metric space $(\mathcal{X},\, d)$ is said to be **well-behaved** if there exists a sequence $(\mathcal{X}_h)_{h\geq 0}$ of subsets of $\mathcal{X}$ satisfying the following properties:*

1. *There exists $N \in \mathbb{N}$ such that for each $h \geq 0$, the set $\mathcal{X}_h$ has N^h elements. We write $\mathcal{X}_h = \{x_{h,i} \colon 1 \leq i \leq N^h\}$ and to each element $x_{h,i}$ is associated a cell $X_{h,i} = \{x \in \mathcal{X} \colon \forall j \neq i \colon d(x, x_{h,i}) \leq d(x, x_{h,j})\}$.*
2. *For all $h \geq 0$ and $1 \leq i \leq N^h$, we have $X_{h,i} = \bigcup_{j=N(i-1)+1}^{Ni} X_{h+1,j}$. The nodes $x_{h+1,j}$ for $N(i-1)+1 \leq j \leq Ni$ are called the children of $x_{h,i}$, which in turn is referred to as the parent of these nodes. We write $p(x_{h+1,j}) = x_{h,i}$ for every $N(i-1)+1 \leq j \leq Ni$.*
3. *We assume that the cells have geometrically decaying radii, i.e., there exist $0 < \rho < 1$ and $0 < v_2 \leq 1 \leq v_1$ such that we have $B(x_{h,i},\, v_2\rho^h) \subseteq X_{h,i} \subseteq B(x_{h,i},\, v_1\rho^h)$ for every $h \geq 0$. Note that we have $2v_2\rho^h \leq \operatorname{diam}(X_{h,i}) \leq 2v_1\rho^h$, where $\operatorname{diam}(X_{h,i}) = \sup_{x,y\in X_{h,i}} d(x,y)$.*

The first property implies that, for every $h \geq 0$, the cells $X_{h,i}$, $1 \leq i \leq N^h$ partition $\mathcal{X}$. This can be observed trivially by *reductio ad absurdum.* The second property intuitively means that, as h grows, we get a more refined partition. The third property implies that the nodes $x_{h,i}$ are evenly spread out in the space.

Additionally, we make use of a notion of dimensionality intrinsic to the design space, namely, the *metric dimension.*

Definition 7. *(Packing, Covering and Metric Dimension (Shekhar et al., 2018)) Let $r \geq 0$.*

- *A subset $\mathcal{X}_1$ of $\mathcal{X}$ is called an r-**packing** of $\mathcal{X}$ if for every $x, y \in \mathcal{X}_1$ such that $x \neq y$, we have $d(x, y) > r$. The largest cardinality of such a set is called the r-packing number of $\mathcal{X}$ with respect to d, and is denoted by $M(\mathcal{X}, r, d)$.*
- *A subset $\mathcal{X}_2$ of $\mathcal{X}$ is called an r-**covering**[2] of $\mathcal{X}$ if for every $x \in \mathcal{X}$, there exists $y \in \mathcal{X}_2$ such that $d(x, y) \leq r$. The smallest cardinality of such a set is called the r-covering number of $\mathcal{X}$ with respect to d, and is denoted by $N(\mathcal{X}, r, d)$.*

[2]Not to be confused with $\boldsymbol{\epsilon}$-covering in Definition 4, where $\boldsymbol{\epsilon} \in \mathbb{R}_+^m$. The meaning will be clear from the context.

- *The **metric dimension** D_1 of $(\mathcal{X}, d)$ is defined as $D_1 = \inf\{a \geq 0 \colon \exists C \geq 0, \forall r > 0 : \log(N(\mathcal{X}, r, d)) \leq C - a\log(r)\}$.*

This dimension coincides with the usual dimension of the space when $\mathcal{X}$ is a subspace of a finite-dimensional Euclidean space. We will upper bound the sample complexity of the algorithm using the metric dimension of $(\mathcal{X}, d)$.

3.3 Prior knowledge on $\boldsymbol{f}$

We model the vector $\boldsymbol{f} = [f^1, \ldots, f^m]^\mathsf{T}$ of objective functions as a realization of an m-output GP with zero mean, i.e., $\boldsymbol{\mu}(x) = 0$ for all $x \in \mathcal{X}$, and some positive definite covariance function $\boldsymbol{k}$.

Definition 8. *An m-**output GP** with index set $\mathcal{X}$ is a collection $(\boldsymbol{f}(x))_{x\in\mathcal{X}}$ of m-dimensional random vectors which satisfies the property that $(\boldsymbol{f}(x_1), \ldots, \boldsymbol{f}(x_n))$ is a Gaussian random vector for all $\{x_1, \ldots, x_n\} \subseteq \mathcal{X}$ and $n \in \mathbb{N}$. The probability law of an m-output GP $(\boldsymbol{f}(x))_{x\in\mathcal{X}}$ is uniquely specified by its (vector-valued) mean function $x \mapsto \boldsymbol{\mu}(x) = \mathbb{E}[\boldsymbol{f}(x)] \in \mathbb{R}^m$ and its (matrix-valued) covariance function $(x_1, x_2) \mapsto \boldsymbol{k}(x_1, x_2) = \mathbb{E}[(\boldsymbol{f}(x_1) - \boldsymbol{\mu}(x_1))(\boldsymbol{f}(x_2) - \boldsymbol{\mu}(x_2))^\mathsf{T}] \in \mathbb{R}^{m\times m}$.*

Functions generated from a GP naturally satisfy smoothness conditions which are very useful while working with metric spaces, as indicated by the following remark.

Remark 1. *Let g be a zero-mean, single-output GP with index set $\mathcal{X}$ and covariance function k. The metric l induced by the GP on $\mathcal{X}$ is defined as $l(x_1, x_2) = \left(\mathbb{E}[(g(x_1) - g(x_2))^2]\right)^{1/2} = (k(x_1, x_1) + k(x_2, x_2) - 2k(x_1, x_2))^{1/2}$. This gives us the following tail bound for $x_1, x_2 \in \mathcal{X}$, and $a \geq 0$: $\mathbb{P}(|g(x_1) - g(x_2)| \geq a) \leq 2\exp\left(-a^2/(2l^2(x_1, x_2))\right)$.*

Let us fix an integer $T \geq 1$. We consider a finite sequence $\tilde{x}_{[T]} = [\tilde{x}_1, \ldots, \tilde{x}_T]^\mathsf{T}$ of designs with the corresponding vector $\boldsymbol{f}_{[T]} = [\boldsymbol{f}(\tilde{x}_1)^\mathsf{T}, \ldots, \boldsymbol{f}(\tilde{x}_T)^\mathsf{T}]^\mathsf{T}$ of unobserved objective values and the (mT-dimensional) vector $\boldsymbol{y}_{[T]} = [\boldsymbol{y}_1^\mathsf{T}, \ldots, \boldsymbol{y}_T^\mathsf{T}]^\mathsf{T}$ of observations, where

$$\boldsymbol{y}_\tau = \boldsymbol{f}(\tilde{x}_\tau) + \boldsymbol{\kappa}_\tau$$

is the observation that corresponds to $\tilde{x}_\tau$ and $\boldsymbol{\kappa}_\tau = [\kappa_\tau^1, \ldots, \kappa_\tau^m]^\mathsf{T}$ is the noise vector that corresponds to this particular evaluation for each $\tau \in [T]$.

The posterior distribution of $\boldsymbol{f}$ given $\boldsymbol{y}_{[T]}$ is that of an m-output GP with mean function $\boldsymbol{\mu}_T$ and covariance function $\boldsymbol{k}_T$ given by

$$\boldsymbol{\mu}_T(x) = \boldsymbol{k}_{[T]}(x)(\boldsymbol{K}_{[T]} + \boldsymbol{\Sigma}_{[T]})^{-1}\boldsymbol{y}_{[T]}^\mathsf{T}$$

and

$$\boldsymbol{k}_T(x, x') = \boldsymbol{k}(x, x') - \boldsymbol{k}_{[T]}(x)(\boldsymbol{K}_{[T]} + \boldsymbol{\Sigma}_{[T]})^{-1}\boldsymbol{k}_{[T]}(x')^\mathsf{T}$$

for all $x, x' \in \mathcal{X}$, where $\boldsymbol{k}_{[T]}(x) = [\boldsymbol{k}(x, \tilde{x}_1), \ldots, \boldsymbol{k}(x, \tilde{x}_T)] \in \mathbb{R}^{m\times mT}$,

$$\boldsymbol{K}_{[T]} = \begin{bmatrix} \boldsymbol{k}(\tilde{x}_1, \tilde{x}_1), & \ldots, & \boldsymbol{k}(\tilde{x}_1, \tilde{x}_T) \\ \vdots & & \vdots \\ \boldsymbol{k}(\tilde{x}_T, \tilde{x}_1), & \ldots, & \boldsymbol{k}(\tilde{x}_T, \tilde{x}_T) \end{bmatrix}, \boldsymbol{\Sigma}_{[T]} = \begin{bmatrix} \sigma^2\boldsymbol{I}_m, & \boldsymbol{0}_m, & \ldots, & \boldsymbol{0}_m \\ \vdots & & & \vdots \\ \boldsymbol{0}_m, & \boldsymbol{0}_m, & \ldots, & \sigma^2\boldsymbol{I}_m \end{bmatrix} \in \mathbb{R}^{mT\times mT},$$

$\boldsymbol{I}_m$ denotes the $m \times m$-dimensional identity matrix, and $\boldsymbol{0}_m$ is the $m \times m$-dimensional zero matrix. Note that this posterior distribution captures the uncertainty in $\boldsymbol{f}(x)$ for all $x \in \mathcal{X}$. In particular, the posterior distribution of $\boldsymbol{f}(x)$ is $\mathcal{N}(\boldsymbol{\mu}_T(x), \boldsymbol{k}_T(x, x))$; and for each $j \in [m]$, the posterior distribution of $f^j(x)$ is $\mathcal{N}(\mu_T^j(x), (\sigma_T^j(x))^2)$, where $(\sigma_T^j(x))^2 = k_T^{jj}(x, x)$. Moreover, the distribution of the corresponding observation $\boldsymbol{y}$ is $\mathcal{N}(\boldsymbol{\mu}_T(x), \boldsymbol{k}_T(x, x) + \sigma^2\boldsymbol{I}_m)$.

3.4 Information gain

Since we aim at finding an $\boldsymbol{\epsilon}$-accurate Pareto set in as few evaluations as possible, we need to learn the most informative designs. In order to do that, we will make use of the notion of *information gain.* Our sample complexity result in Theorem 1 depends on the maximum information gain.

In Bayesian experimental design, the informativeness of a finite sequence $\tilde{x}_{[T]}$ of designs is quantified by $I(\boldsymbol{y}_{[T]};\boldsymbol{f}_{[T]}) = H(\boldsymbol{y}_{[T]}) - H(\boldsymbol{y}_{[T]}|\boldsymbol{f}_{[T]})$, where $H(\cdot)$ denotes the entropy of a random vector and $H(\cdot|\boldsymbol{f}_{[T]})$ denotes the conditional entropy of a random vector with respect to $\boldsymbol{f}_{[T]}$. This measure is called the information gain, which gives us the decrease of entropy of $\boldsymbol{f}_{[T]}$ given the observations $\boldsymbol{y}_{[T]}$. We define the *maximum information gain* as $\gamma_T = \max_{\boldsymbol{y}_{[T]}} I(\boldsymbol{y}_{[T]};\boldsymbol{f}_{[T]})$.

4 Adaptive ϵ-PAL algorithm

In this section, we introduce our algorithm, *Adaptive ϵ-Pareto Active Learning* (PAL), which builds on the ϵ-PAL algorithm of Zuluaga et al. (2016) and utilizes adaptive discretization techniques to enable an efficient navigation of the continuous search space. The algorithm is largely technical, thus we divide the description of each of its phases into more comprehensible subsections.

The system operates in rounds $t \geq 1$. In each round t, the algorithm picks a design $x_t \in \mathcal{X}$, and assuming that it already had τ evaluations, it subsequently decides whether or not to obtain the $\tau+1$st noisy observation $\boldsymbol{y}_{\tau+1} = [y^1_\tau, \ldots, y^m_{\tau+1}]^\mathsf{T}$ of the latent function $\boldsymbol{f}$ at x_t. At the end, our algorithm returns a subset $\hat{P}$ of $\mathcal{X}$ which is guaranteed to be an $\boldsymbol{\epsilon}$-accurate Pareto set with high probability and the associated set $\hat{\mathcal{P}}$ of nodes which we will define later. The pseudocode is given in Algorithm 1.

4.1 Modeling

We maintain two sets of time indices, one counting the total number of iterations, denoted by t, and the other counting only the evaluation rounds, denoted by τ. The algorithm evaluates a design only in some rounds. For this reason, we also define the following auxiliary time variables which help us understand the chronological connection between the values of t and τ. We let τ_t represent the number of evaluations before round $t \geq 1$ and let t_τ denote the round when evaluation $\tau \geq 0$ is made, with the convention $t_0 = 0$. The sequence $(t_\tau)_{\tau\geq 0}$ is an increasing sequence of stopping times. Note that we have $\tau_{t_\tau+1} = \tau$ for each $\tau \in \mathbb{N}$.

Iteration over individual designs may not be feasible when the cardinality of the design space is very large. Thus, we consider partitioning the space into regions of similar designs, i.e., two designs in $\mathcal{X}$ which are at a close distance have similar outcomes in each objective. This is a natural property of GP-sampled functions as discussed in Remark 1. Since our metric space is well-behaved (see Definition 6), there exist an $N \in \mathbb{N}$ and a sequence $(\mathcal{X}_h)_{h\geq 0}$ of subsets of $\mathcal{X}$ such that for each $h \geq 0$, the set $\mathcal{X}_h$ contains N^h nodes denoted by $x_{h,1}, \ldots, x_{h,N^h}$. For each $i \in [N^h]$, the associated cell of node $x_{h,i}$ is denoted by $X_{h,i}$.

At each round $t \in \mathbb{N}$, the algorithm maintains a set $\mathcal{S}_t$ of *undecided nodes* and a set $\mathcal{P}_t$ of *decided nodes.* For an undecided node in $\mathcal{S}_t$, its associated cell consists of designs for which we are undecided about including in the $\boldsymbol{\epsilon}$-accurate Pareto set. Similarly, for a decided node in $\mathcal{P}_t$, its associated cell consists of designs that we decide to include in the $\boldsymbol{\epsilon}$-accurate Pareto set. At the beginning of round $t = 1$ (initialization), we set $\mathcal{S}_1 = \{x_{0,1}\}$ and $\mathcal{P}_1 = \emptyset$. Within each round $t \in \mathbb{N}$, the sets $\mathcal{P}_t$ and $\mathcal{S}_t$ are updated during the discarding, $\boldsymbol{\epsilon}$-covering and refining/evaluating phases of the round; at the end of round t, their finalized contents are set as $\mathcal{P}_{t+1}$ and $\mathcal{S}_{t+1}$, respectively, as a preparation for round $t+1$. For each $t \in \mathbb{N}$, the algorithm performs round t as long as $\mathcal{S}_t \neq \emptyset$ at the beginning of round t; otherwise, it terminates and returns $\hat{\mathcal{P}} = \mathcal{P}_t$.

In addition to the sets of undecided and decided nodes, the algorithm maintains a set $\mathcal{A}_t$ of *active nodes*, which is defined as the union $\mathcal{S}_t \cup \mathcal{P}_t$ at the beginning of each round $t \in \mathbb{N}$. While the sets $\mathcal{S}_t$ and $\mathcal{P}_t$ are updated within round t as described above, the set $\mathcal{A}_t$ is kept fixed throughout the round with its initial content. Note that $\mathcal{A}_1 = \{x_{0,1}\}$.

At round $t \in \mathbb{N}$, the algorithm considers each node $x_{h,i} \in \mathcal{A}_t$. Let $j \in [m]$. We define the *lower index* of $x_{h,i}$ in the jth objective as $L^j_t(x_{h,i}) = \underline{B}^j_t(x_{h,i}) - V_h$, where $\underline{B}^j_t(x_{h,i})$ is a high probability lower bound on the jth objective value at $x_{h,i}$ and is defined as

$$\underline{B}_t^j(x_{h,i}) = \max\{\mu_{\tau_t}^j(x_{h,i}) - \beta_{\tau_t}^{1/2}\sigma_{\tau_t}^j(x_{h,i}), \mu_{\tau_t}^j(p(x_{h,i})) - \beta_{\tau_t}^{1/2}\sigma_{\tau_t}^j(p(x_{h,i})) - V_{h-1}\} \ .$$

Here, β_τ is a parameter to be defined later and V_h is a high probability upper bound on the maximum variation of the objective j inside region $X_{h,i}$, which will also be defined later. Similarly, we define the *upper index* of $x_{h,i}$ in the jth objective as $U_t^j(x_{h,i}) = \bar{B}_t^j(x_{h,i}) + V_h$, where $\bar{B}_t^j(x_{h,i})$ is a high probability upper bound on the jth objective value at $x_{h,i}$ and is defined as

$$\bar{B}_t^j(x_{h,i}) = \min\{\mu_{\tau_t}^j(x_{h,i}) + \beta_{\tau_t}^{1/2}\sigma_{\tau_t}^j(x_{h,i}), \mu_{\tau_t}^j(p(x_{h,i})) + \beta_{\tau_t}^{1/2}\sigma_{\tau_t}^j(p(x_{h,i})) + V_{h-1}\} \ .$$

We denote by $\boldsymbol{L}_t(x_{h,i}) = [L_t^1(x_{h,i}), \ldots, L_t^m(x_{h,i})]^\mathsf{T}$ the *lower index vector* of the node $x_{h,i}$ at round t and similarly by $\boldsymbol{U}_t(x_{h,i}) = [U_t^1(x_{h,i}), \ldots, U_t^m(x_{h,i})]^\mathsf{T}$ the corresponding *upper index vector*. We also let $\boldsymbol{V}_h$ denote the m-dimensional vector with all entries being equal to V_h. Next, we define the *confidence hyper-rectangle* of node $x_{h,i}$ at round t as

$$\boldsymbol{Q}_t(x_{h,i}) = \{\boldsymbol{y} \in \mathbb{R}^m : \boldsymbol{L}_t(x_{h,i}) \preceq \boldsymbol{y} \preceq \boldsymbol{U}_t(x_{h,i})\} \ ,$$

which captures the uncertainty in the learner's prediction of the objective values. Then, the posterior mean vector $\boldsymbol{\mu}_{\tau_t}(x_{h,i}) = [\mu_{\tau_t}^1(x_{h,i}), \ldots, \mu_{\tau_t}^m(x_{h,i})]^\mathsf{T}$ and the variance vector $\boldsymbol{\sigma}_{\tau_t}(x_{h,i}) = [\sigma_{\tau_t}^1(x_{h,i}), \ldots, \sigma_{\tau_t}^m(x_{h,i})]^\mathsf{T}$ are computed by using the GP inference outlined in Section 3. We define the *cumulative confidence hyper-rectangle* of $x_{h,i}$ at round t as

$$\boldsymbol{R}_t(x_{h,i}) = \boldsymbol{R}_{t-1}(x_{h,i}) \cap \boldsymbol{Q}_t(x_{h,i}) \tag{1}$$

assuming that $\boldsymbol{R}_{t-1}(x_{h,i})$ is well-defined at round $t-1$ (the case $t \geq 2$) or using the convention that $\boldsymbol{R}_0(x_{0,1}) = \mathbb{R}^m$ since $\mathcal{A}_1 = \{x_{0,1}\}$ (the case $t = 1$). The well-definedness assumption will be verified in the refining/evaluating phase below.

4.2 Discarding phase

In order to correctly identify designs to be discarded under uncertainty, we need to compare the pessimistic and optimistic outcomes of designs. First, we define dominance under uncertainty.

Definition 9. *Let $t \in \mathbb{N}$ and let $x, y \in \mathcal{A}_t$ be two nodes with $x \neq y$. We say that x is **$\boldsymbol{\epsilon}$-dominated by y under uncertainty** at round t if $\max(\boldsymbol{R}_t(x)) \preceq_{\boldsymbol{\epsilon}} \min(\boldsymbol{R}_t(y))$, where we define $\max(\boldsymbol{R}_t(x))$ as the unique vector $\boldsymbol{v} \in \boldsymbol{R}_t(x)$ such that $v^j \geq z^j$ for every $j \in [m]$ and $\boldsymbol{z} = (z^1, \ldots, z^j) \in \boldsymbol{R}_t(x)$, and we define $\min(\boldsymbol{R}_t(y))$ in a similar fashion.*

If a node $x \in \mathcal{A}_t$ is $\boldsymbol{\epsilon}$-dominated by any other node in $\mathcal{A}_t$ under uncertainty, then the algorithm is confident enough to discard it. To check this, the algorithm compares x with all of the *pessimistic* available points as introduced next.

Definition 10. ***(Pessimistic Pareto set)*** *Let $t \geq 1$ and let $D \subseteq \mathcal{A}_t$ be a set of nodes. We define $p_{pess,t}(D)$, called the **pessimistic Pareto set** of D at round t, as the set of all nodes $x \in D$ for which there is no other node $y \in D \setminus \{x\}$ such that $\min(\boldsymbol{R}_t(x)) \preceq \min(\boldsymbol{R}_t(y))$. We call a design in $p_{pess,t}(D)$ a pessimistic Pareto design of D at round t.*

Here we are interested in finding the nodes, say x, which are Pareto optimal in the most pessimistic scenario when their objective values turn out to be $\min \boldsymbol{R}_t(x)$. We do this in order to identify which nodes (and their associated cells) to discard with overwhelming probability. More precisely, the algorithm calculates $\mathcal{P}_{pess,t} = p_{pess,t}(\mathcal{A}_t)$ first. For each $x_{h,i} \in \mathcal{S}_t \backslash \mathcal{P}_{pess,t}$, it checks if $\max(\boldsymbol{R}_t(x_{h,i})) \preceq_{\boldsymbol{\epsilon}} \min(\boldsymbol{R}_t(x))$ for some $x \in \mathcal{P}_{pess,t}$. In this case, node $x_{h,i}$ is discarded, that is, it is removed from $\mathcal{S}_t$, and will not be considered in the rest of the algorithm; otherwise no change is made in $\mathcal{S}_t$.

4.3 ϵ-Covering phase

The overall aim of the learner is to empty the set $\mathcal{S}_t$ of undecided nodes as fast as possible. A node $x_{h,i} \in \mathcal{S}_t$ is moved to the decided set $\mathcal{P}_t$ if it is determined that the associated cell $X_{h,i}$ belongs to an $\boldsymbol{\epsilon}$-accurate Pareto set $\mathcal{O}_\epsilon$ with high probability. To check this, the notion in the next definition is useful. Let us denote by $\mathcal{W}_t$ the union $\mathcal{P}_t \cup \mathcal{S}_t$ at the end of the discarding phase. Note that $\mathcal{W}_t \subseteq \mathcal{A}_t$ but the two sets do not coincide in general due to the discarding phase.

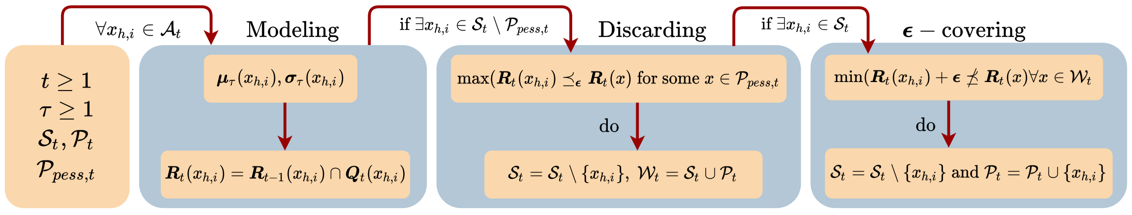
$\forall x_{h,i} \in \mathcal{A}_t$
Modeling
if $\exists x_{h,i} \in \mathcal{S}_t \setminus \mathcal{P}_{pess,t}$
Discarding
if $\exists x_{h,i} \in \mathcal{S}_t$
$\boldsymbol{\epsilon}$ − covering
$t \geq 1$
$\tau \geq 1$
$\mathcal{S}_t, \mathcal{P}_t$
$\mathcal{P}_{pess,t}$
$\boldsymbol{\mu}_\tau(x_{h,i}), \boldsymbol{\sigma}_\tau(x_{h,i})$
$\boldsymbol{R}_t(x_{h,i}) = \boldsymbol{R}_{t-1}(x_{h,i}) \cap \boldsymbol{Q}_t(x_{h,i})$
$\max(\boldsymbol{R}_t(x_{h,i})) \preceq_{\boldsymbol{\epsilon}} \boldsymbol{R}_t(x)$ for some $x \in \mathcal{P}_{pess,t}$
do
$\mathcal{S}_t = \mathcal{S}_t \setminus \{x_{h,i}\}$, $\mathcal{W}_t = \mathcal{S}_t \cup \mathcal{P}_t$
$\min(\boldsymbol{R}_t(x_{h,i})) + \boldsymbol{\epsilon} \not\preceq \boldsymbol{R}_t(x) \forall x \in \mathcal{W}_t$
do
$\mathcal{S}_t = \mathcal{S}_t \setminus \{x_{h,i}\}$ and $\mathcal{P}_t = \mathcal{P}_t \cup \{x_{h,i}\}$

Figure 1: This is an illustration of the modeling, discarding, and $\boldsymbol{\epsilon}$-covering phases.

Definition 11. *Let $x_{h,i} \in \mathcal{S}_t$. We say that the cell $X_{h,i}$ associated to node $x_{h,i}$* ***belongs to an $\mathcal{O}_\epsilon$ with high probability*** *if there is no $x \in \mathcal{W}_t$ such that $\min(\boldsymbol{R}_t(x_{h,i})) + \boldsymbol{\epsilon} \preceq \max(\boldsymbol{R}_t(x))$.*

For each $x_{h,i} \in \mathcal{S}_t$, the algorithm checks if $X_{h,i}$ belongs to an $\mathcal{O}_\epsilon$ with high probability in view of Definition 11. In this case, $x_{h,i}$ is removed from the set $\mathcal{S}_t$ of undecided nodes and is moved to the set $\mathcal{P}_t$ of decided nodes; otherwise, no change is made. The nodes in $\mathcal{P}_t$ are never removed from this set; hence, they will be returned by the algorithm as part of the set $\hat{\mathcal{P}}$ at termination. In the appendix, we show that the union $\hat{P} = \bigcup_{x_{h,i} \in \hat{\mathcal{P}}} X_{h,i}$ of the cells is an $\boldsymbol{\epsilon}$-accurate Pareto cover, according to Definition 5, with high probability.

Note that while the sets $\mathcal{S}_t, \mathcal{P}_t$ can be modified during this phase, the set $\mathcal{W}_t$ does not change.

The modeling, discarding, and $\boldsymbol{\epsilon}$-covering phases of the algorithm are illustrated in Figure 1.

4.4 Refining/evaluating phase

While $\mathcal{S}_t \neq \emptyset$, the algorithm selects a design $x_t = x_{h_t,i_t} \in \mathcal{W}_t$ that corresponds to a node with depth h_t and index i_t, according to the following rule. First, for a given node $x_{h,i} \in \mathcal{W}_t$, we define

$$\omega_t(x_{h,i}) = \max_{y,y' \in \boldsymbol{R}_t(x_{h,i})} \|y - y'\|_2 \ , \tag{2}$$

which is the diameter of its cumulative confidence hyper-rectangle in $\mathbb{R}^m$. The algorithm picks the most uncertain node for evaluation in order to decrease uncertainty. Hence, among the available points in $\mathcal{W}_t$, the node x_{h_t,i_t} with the maximum such diameter is chosen by the algorithm. We denote the diameter of the cumulative confidence hyper-rectangle associated with the selected node by $\overline{\omega}_t$ and formally define it as $\overline{\omega}_t = \max_{x_{h,i} \in \mathcal{W}_t} \omega_t(x_{h,i})$. Since the learner is not sure about discarding x_{h_t,i_t} or moving it to $\mathcal{P}_t$, he decides whether to refine the associated region X_{h_t,i_t} or evaluating the objective function at the current node based on the following rule.

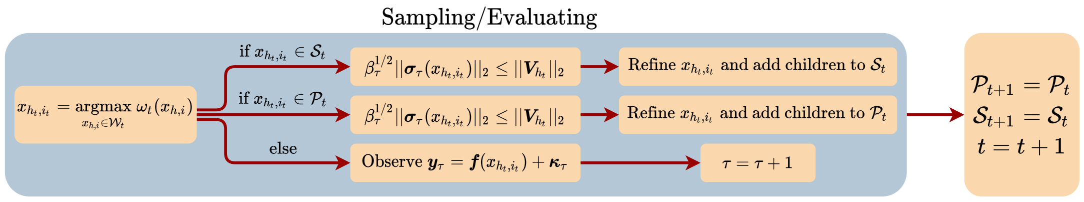
Sampling/Evaluating
$x_{h_t,i_t} = \underset{x_{h,i} \in \mathcal{W}_t}{\operatorname{argmax}}\ \omega_t(x_{h,i})$
if $x_{h_t,i_t} \in \mathcal{S}_t$
if $x_{h_t,i_t} \in \mathcal{P}_t$
else
$\beta_\tau^{1/2} ||\boldsymbol{\sigma}_\tau(x_{h_t,i_t})||_2 \leq ||\boldsymbol{V}_{h_t}||_2$
$\beta_\tau^{1/2} ||\boldsymbol{\sigma}_\tau(x_{h_t,i_t})||_2 \leq ||\boldsymbol{V}_{h_t}||_2$
Observe $\boldsymbol{y}_\tau = \boldsymbol{f}(x_{h_t,i_t}) + \boldsymbol{\kappa}_\tau$
Refine x_{h_t,i_t} and add children to $\mathcal{S}_t$
Refine x_{h_t,i_t} and add children to $\mathcal{P}_t$
$\tau = \tau + 1$
$\mathcal{P}_{t+1} = \mathcal{P}_t$
$\mathcal{S}_{t+1} = \mathcal{S}_t$
$t = t + 1$

Figure 2: This is an illustration of the sampling and refining phase.

- *Refine:* If $\beta_{\tau_t}^{1/2} \|\boldsymbol{\sigma}_{\tau_t}(x_{h_t,i_t})\|_2 \leq \|\boldsymbol{V}_{h_t}\|_2$, then x_{h_t,i_t} is expanded, i.e., the N children nodes $\{x_{h_t+1,j} : N(i_t - 1) + 1 \leq j \leq i_t\}$ of x_{h_t,i_t} are generated. If $x_{h_t,i_t} \in \mathcal{S}_t$, then these newly generated nodes are added to $\mathcal{S}_t$ while x_{h_t,i_t} is removed from $\mathcal{S}_t$. An analogous operation is performed if $x_{h_t,i_t} \in \mathcal{P}_t$. In each case, for each j with $N(i_t - 1) + 1 \leq j \leq i_t$, the newly generated node $x_{h_t+1,j}$ inherits the cumulative confidence hyper-rectange of its parent node $x_{h_t,i_t} \in \mathcal{A}_t$ as calculated by equation 1, that is, we define $\boldsymbol{R}_t(x_{h_t+1,j}) = \boldsymbol{R}_t(x_{h_t,i_t}) = \boldsymbol{R}_{t-1}(x_{h_t,i_t}) \cap \boldsymbol{Q}_t(x_{h_t,i_t})$. This way, for every node $x \in \mathcal{P}_t \cup \mathcal{S}_t$ at the end of refining, the cumulative confidence hyper-rectangles up to round t are well-defined and we have $\boldsymbol{R}_0(x) \supseteq \boldsymbol{R}_1(x) \supseteq \ldots \supseteq \boldsymbol{R}_t(x)$. In particular, the well-definedness assumption for equation 1 is verified for round $t+1$ since $\mathcal{A}_{t+1}$ is defined as $\mathcal{P}_t \cup \mathcal{S}_t$ at the end of this phase.

- *Evaluate:* If $\beta_{\tau_t}^{1/2} \left\|\boldsymbol{\sigma}_{\tau_t}(x_{h_t,i_t})\right\|_2 > \|\boldsymbol{V}_{h_t}\|_2$, then the objective function is evaluated at the point x_{h_t,i_t}, i.e., we observe the noisy sample $\boldsymbol{y}_{\tau_t}$ and update the posterior statistics of x_{h_t,i_t}. No change is made in $\mathcal{S}_t$ and $\mathcal{P}_t$.

This phase of the algorithm is illustrated in Figure 2. The evolution of the partitioning of the design space is illustrated in Figure 3.

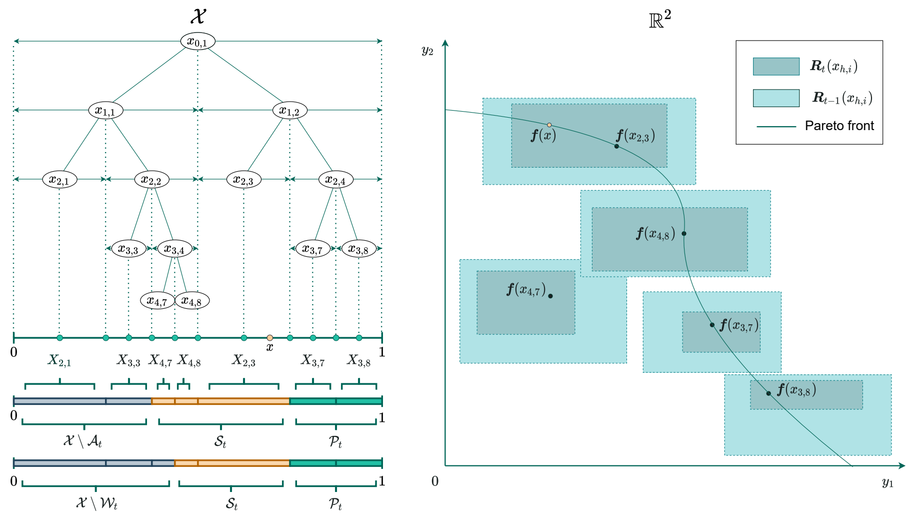

Figure 3: This is an illustration of the structural way of partitioning the design space. In this example we take $\mathcal{X} = [0,1]$, $m = 2$ and $\boldsymbol{\epsilon} = \boldsymbol{0}$. On the left, we can see the partition of $\mathcal{X}$ at the beginning of round t. Note that $\mathcal{S}_t = \{x_{4,7}, x_{4,8}, x_{2,3}\}$ and $\mathcal{P}_t = \{x_{3,7}, x_{3,8}\}$, while $x_{2,1}$ and $x_{3,3}$ have been discarded in some prior round. On the right, we see the corresponding confidence hyper-rectangles of these nodes. At the beginning of round t, prior to the modeling phase, the hyper-rectangle of node $x_{h,i}$ is $\boldsymbol{R}_{t-1}(x_{h,i})$. Note that, in the discarding phase, node $x_{4,8}$ will be discarded since $\max(\boldsymbol{R}_t(x_{4,8})) \preceq_{\boldsymbol{\epsilon}} \min(\boldsymbol{R}_t(x_{4,7}))$. Thus, $x_{4,7}$ will be removed from $\mathcal{S}_t$ by the end of the phase. Furthermore, note that more than one Pareto optimal designs take values in one hyper-rectangle. This is because the node of a region containing Pareto optimal points is not necessarily one of these points. For example, we have $x \in X_{2,3}$ and $\boldsymbol{f}(x) \in \boldsymbol{R}_t(x_{2,3})$.

4.5 Termination

If $\mathcal{S}_t = \emptyset$ at the beginning of round t, then the algorithm terminates. We show in the appendix that at the latest, the algorithm terminates when $\overline{\omega}_t \leq \min_j \epsilon^j$. Upon termination, it returns a non-empty set $\hat{\mathcal{P}}$ of decided nodes together with the corresponding $\boldsymbol{\epsilon}$-accurate Pareto set $\hat{P} = \bigcup_{x_{h,i} \in \hat{\mathcal{P}}} X_{h,i}$, which is the union of the cells corresponding to the nodes in $\hat{\mathcal{P}}$.

5 Sample complexity bounds

We state the main result in this section. Its proof is composed of a sequence of lemmas which are given in the appendix. We first state the necessary assumptions (Assumption 1) on the metric and the kernel under which the result holds. Then, we provide a sketch of its proof and subsequently give an example of an m-output GP for which the maximum information gain is linear in T.

Assumption 1. *The class $\mathcal{K}$ of covariance functions to which we restrict our focus satisfies the following criteria for any $\boldsymbol{k} \in \mathcal{K}$: (1) For any $x, y \in \mathcal{X}$ and $j \in [m]$, we have $l_j(x, y) \leq C_{\boldsymbol{k}} d(x, y)^\alpha$, for suitable $C_{\boldsymbol{k}} > 0$ and $0 < \alpha \leq 1$. Here, l_j is the natural metric induced on $\mathcal{X}$ by the jth component of the GP in Definition 8 as given in Remark 1 with covariance function k^{jj}. (2) We assume bounded variance, that is, for any $x \in \mathcal{X}$ and $j \in [m]$, we have $k^{jj}(x, x) \leq 1$.*

Theorem 1. *Let $\boldsymbol{\epsilon} = [\epsilon^1, \ldots, \epsilon^m]^\mathsf{T}$ be given with $\epsilon = \min_{j \in [m]} \epsilon^j > 0$. Let $\delta \in (0, 1)$ and $\bar{D} > D_1$. For each $h \geq 0$, let $V_h \in \tilde{O}(\rho^{\alpha h})$; for each $\tau \in \mathbb{N}$, let $\beta_\tau \in O(\log(\tau^2/\delta))$; the exact definitions are given in the appendix. When we run Adaptive $\boldsymbol{\epsilon}$-PAL with prior $GP(0, \boldsymbol{k})$ and noise $\mathcal{N}(0, \sigma^2)$, the following holds with probability at least $1 - \delta$.*

An $\boldsymbol{\epsilon}$-accurate Pareto set can be found with at most T function evaluations, where T is the smallest natural number satisfying

$$\min \left\{ K_1 \beta_T T^{\frac{-\alpha}{\bar{D}+2\alpha}} (\log T)^{\frac{-(\bar{D}+\alpha)}{\bar{D}+2\alpha}} + K_2 T^{\frac{-\alpha}{\bar{D}+2\alpha}} (\log T)^{\frac{\alpha}{\bar{D}+2\alpha}} , \sqrt{\frac{C \beta_T \gamma_T}{T}} \right\} \leq \epsilon \ ,$$

where C and K_1 are constants that are defined in the appendix and do not depend on T, K_2 is logarithmic in T, and γ_T is the maximum information gain which depends on the choice of $\boldsymbol{k}$.

Sketch of the proof of Theorem 1: We divide the proof of Theorem 1 into three essential parts. First, we prove that the algorithm terminates in finite time, and examine the events that necessitate and the ones that follow termination. Here, it is important to note that if the largest uncertainty diameter in a given round t is less than or equal to ϵ, then the algorithm terminates. This introduces a way on how to proceed on upper bounding sample complexity, namely, we can upper bound the sum of these uncertainty diameters over all rounds. Then, by observing that i) the term V_h decays to 0 as h grows indefinitely, which means that the algorithm refines up until a finite number of tree levels, and that ii) a node cannot be refined more than a finite number of times before expansion, we can conclude that the algorithm terminates in finite time. After this point we prove, using Hoeffding bounds, that the true function values live inside the uncertainty hyper-rectangles with high probability. We then proceed on first proving the dimension-type sample complexity bounds and then the information-type bounds. We use the aforementioned termination condition and upper bound the sum of the uncertainty diameters over all rounds. We use Cauchy-Schwarz inequality to manipulate the expression as we desire and then upper bound it in terms of information gain using an already established result in the paper. For the dimension-type bounds, we consider expressing the sum over rounds as a sum over levels h of the tree of partitions and upper bound it using the notion of metric dimension. We take the minimum of the two bounds, thus achieving the bound stated in Theorem 1.

Remark 2. *Note that we minimize over two different bounds in Theorem 1. The term that involves γ_T corresponds to the information-type bound, while the other term corresponds to the metric dimension-type bound. Equivalently, we can express our information-type bound as $\tilde{O}(g(\epsilon))$ where $g(\epsilon) = \min\{T \geq 1 : \sqrt{\gamma_T/T} \leq \epsilon\}$ and our metric dimension-type bound as $\tilde{O}(\epsilon^{-(\frac{\bar{D}}{\alpha}+2)})$ for any $\bar{D} > D_1$. For certain kernels, such as squared exponential and Matérn kernels, γ_T can be upper bounded by a sublinear function of T (see Srinivas et al. (2012)). Our information type-bound is of the same form as in Zuluaga et al. (2016). When $\mathcal{X}$ is a finite subset of the Euclidean space, we have $D_1 = 0$, and thus, our metric-dimension type bound becomes near-$O(1/\epsilon^2)$, which is along the same lines with the almost optimal, gap-dependent near-$O(1/gap^2)$ bound for Pareto front identification in Auer et al. (2016).*

In order to prove the bounds in Theorem 1, even for infinite $\mathcal{X}$, we propose a novel way of defining the confidence hyper-rectangles and refining them. Since Adaptive $\boldsymbol{\epsilon}$-PAL discards, $\boldsymbol{\epsilon}$-covers and refines/evaluates in ways different than $\boldsymbol{\epsilon}$-PAL in Zuluaga et al. (2016), we use different arguments in the proof to show when the algorithm converges and what it returns when it converges. In particular, for the information-type bound, we exploit the dependence structure between the objectives. Moreover, having two different bounds allows us to use the best of both, as it is known that for certain kernels, the metric dimension-type bound can be tighter than the information-type bound. That is the implication of our next result. The proof is mainly technical, thus we defer it to the Appendix.

Proposition 1. *There exists a multi-output GP $\boldsymbol{f}$, with covariance function satisfying Assumption 1 and a sequence of T noisy observations made on $\boldsymbol{f}$, such that we have $I(\boldsymbol{y}_{[T]}, \boldsymbol{f}_{[T]}) \geq \Omega(T)$.*

6 Computational complexity analysis

The most significant subroutines of Adaptive $\boldsymbol{\epsilon}$-PAL in terms of computational complexity are the modeling, discarding, and $\boldsymbol{\epsilon}$-covering phases. Below, we inspect the computational complexity of these three phases separately.

Modeling. In the modeling phase, mean and variance values of the GP surrogate are computed for every node point in the leaf set. This results in a complexity of $\mathcal{O}(\tau^3 + n\tau^2)$, where τ is the number of evaluations and n is the number of node points at a particular round. Note that the bound on n depends on the maximum depth of the search tree.

Discarding. This phase can be further separated into two substeps. In the first substep, the pessimistic Pareto front is determined by choosing the leaf nodes whose lower confidence bounds are not dominated by any other point in the leaf set. In the second phase, the leaf points whose upper confidence bounds are $\boldsymbol{\epsilon}$-dominated by pessimistic Pareto set points are discarded. If done naively by comparing each point with all the other points, both of the phases can result in $\mathcal{O}(n^2)$ complexity. However, there are efficient methods that achieve lower computational complexity. In our implementation, we adopt an efficient algorithm in Kung et al. (1975) to reduce computational complexity to $\mathcal{O}(n \log n)$ when $m = 2$ or $m = 3$. Another algorithm proposed by Kung et al. (1975) can be implemented and it achieves $\mathcal{O}(n(\log n)^{m-2} + n \log n)$ complexity for $m > 3$.

$\boldsymbol{\epsilon}$-Covering. In the $\boldsymbol{\epsilon}$-covering phase, the algorithm moves the nodes that are not dominated by any other node to $\hat{\mathcal{P}}$. Similar to the discarding phase, when implemented naively, the $\boldsymbol{\epsilon}$-covering phase results in $\mathcal{O}(n^2)$ complexity. An adaptation of Kung et al. (1975) algorithm can be implemented as in the case of discarding phase to achieve sub-quadratic complexity which results in $\mathcal{O}(n \log n)$ for $m = 2$ and $m = 3$ and $\mathcal{O}(n(\log n)^{m-2} + n \log n)$ for $m > 3$.

In general, we observed that the number of evaluations was much smaller than the number of nodes throughout a run. Therefore we can say that discarding and ϵ-covering phases are the main bottlenecks in our implementation which scale sub-quadratically with the number of points.

7 Experiments

This section empirically evaluates Adaptive ϵ-PAL, comparing its performance and efficiency against state-of-the-art multi-objective Bayesian optimization (MOBO) methods. We focus on validating the effectiveness of the adaptive discretization strategy and assessing the algorithm's ability to find an $\boldsymbol{\epsilon}$-accurate Pareto set sample-efficiently.

7.1 Performance metrics

We use a combination of three performance metrics: ϵ-accuracy ratio, ϵ-coverage ratio, and average mean-squared error (MSE). Since the objective functions tested are continuous, the true Pareto front contains infinitely many points. To compute the metrics, we approximate the true Pareto front by sampling $10,000$ points uniformly from the input space, evaluating the objective function at these points, and identifying the Pareto optimal designs among them. We ensure the sampling density is high (gap between consecutive samples less than $\epsilon/10$) so that this discretized front serves as a reasonable ground truth.

$\boldsymbol{\epsilon}$-accuracy ratio. This metric measures the quality of the predicted Pareto set $\hat{P}$. It is computed as the ratio of predicted points $\boldsymbol{f}(x)$ (with $x \in \hat{P}$) that fall within the $\boldsymbol{\epsilon}$-Pareto front $\mathcal{Z}_{\boldsymbol{\epsilon}}(\mathcal{X})$ to the total number of points in $\hat{P}$. Operationally, we check if a predicted point is $\boldsymbol{\epsilon}$-dominated by some point on the approximated true Pareto front, or equivalently, if it is at most $2\boldsymbol{\epsilon}$ away (in the sense of $\preceq_{2\boldsymbol{\epsilon}}$) from the closest true Pareto point. While high $\boldsymbol{\epsilon}$-accuracy is desirable, it does not guarantee that the entire Pareto front is well-represented. We also need to assess how well the predicted set covers the true front, which is measured by the ϵ-coverage metric.

$\boldsymbol{\epsilon}$-coverage ratio. This metric measures how well the predicted set $\hat{P}$ covers the true Pareto front. It is computed as the ratio of true Pareto points (from the discretized approximation) that are $\boldsymbol{\epsilon}$-dominated by at least one point in $\boldsymbol{f}(\hat{P})$ to the total number of true Pareto points. Equivalently, we check if a true Pareto point is within $2\boldsymbol{\epsilon}$ (in the sense of $\preceq_{2\boldsymbol{\epsilon}}$) of the closest predicted point.

Average mean-squared error (MSE). This metric complements $\boldsymbol{\epsilon}$-coverage by quantifying the average closeness of the predicted front to the true front. It is computed by averaging the squared Euclidean distance between each true Pareto point (in the objective space) and the closest predicted Pareto point in $\boldsymbol{f}(\hat{P})$. Ideally, we seek high values for both accuracy and coverage, and a low value for MSE.

7.2 Multi-objective Bayesian optimization algorithms

We compare Adaptive $\boldsymbol{\epsilon}$-PAL with three state-of-the-art MOBO methods: PESMO (Hernández-Lobato et al., 2016), ParEGO (Knowles, 2006), and USeMO (Belakaria et al., 2020). Since PESMO, ParEGO, and USeMO require a pre-specified evaluation budget, while Adaptive $\boldsymbol{\epsilon}$-PAL terminates based on its confidence criteria, we set the budget for the competitors to be at least as large as the number of evaluations performed by Adaptive $\boldsymbol{\epsilon}$-PAL upon termination in our experiments. This ensures competitors have access to at least as much information. We used the implementations for PESMO and ParEGO from the Spearmint Bayesian optimization library[3] and for USeMO from the authors' open-source code[4].

For completeness, we also attempted to benchmark against the original ϵ-PAL algorithm (Zuluaga et al., 2016). However, we encountered difficulties achieving termination when running both the authors' publicly available code and our own implementation based on the published pseudocode. As we were unable to obtain completed runs, we have excluded ϵ-PAL from the presented results.

7.3 Simulation setup & results

We first sampled 10 distinct objective functions from the specified GP prior (detailed below). For each of these functions, we then ran each algorithm 5 times, using different random seeds for each run to vary the observation noise and any internal randomness within the algorithms.

Setup. We simulate a problem with a 1-dimensional input space $\mathcal{X} = [0, 1]$ and a 2-dimensional objective space ($m = 2$). The objective function $\boldsymbol{f}$ is sampled from a GP with a zero mean function and independent squared exponential kernels for each objective: $k^1(x, x') = 0.5 \exp(-\frac{(x-x')^2}{2\times 0.1^2})$ and $k^2(x, x') = 0.1 \exp(-\frac{(x-x')^2}{2\times 0.06^2})$. Observation noise is Gaussian with $\sigma = 0.01$ (variance $\sigma^2 = 10^{-4}$).

Adaptive $\boldsymbol{\epsilon}$-PAL. We run Adaptive $\boldsymbol{\epsilon}$-PAL with a target accuracy $\boldsymbol{\epsilon} = (0.05, 0.05)$ and confidence $\delta = 0.05$. The tree parameters were set to $N = 2$, $\rho = 1/2$, and $v_1 = v_2 = 1$. To investigate the impact of the maximum tree depth, we tested three settings for $h_{\max}$. The first setting used the theoretically derived value $h_{\max} = 24$, calculated using Lemma 4 based on the target ϵ. The other two settings used practical, reduced depths of $h_{\max} = 10$ and $h_{\max} = 9$ to evaluate the trade-off between computational cost and performance. When using these practical depth limits, refinement beyond $h_{\max}$ was prevented by setting $V_h = 0$ for all $h \geq h_{\max}$. In this specific experimental run, Adaptive $\boldsymbol{\epsilon}$-PAL terminated after approximately 50 evaluations for $h_{\max} = 24$, 40 evaluations for $h_{\max} = 10$, and 35 evaluations for $h_{\max} = 9$.

PESMO and ParEGO. We run PESMO and ParEGO for 100 iterations (exceeding Adaptive $\boldsymbol{\epsilon}$-PAL's evaluations). We use the same $\delta = 0.05$ where applicable. The acquisition function optimization starts from the best point on a grid of size 1000, followed by L-BFGS optimization.

USeMO. We run USeMO for 100 iterations. Expected Improvement (EI) is used as the acquisition function, aggregated using the Tchebycheff scalarization as recommended by the authors.

Results. We evaluate the predicted Pareto front points returned by each algorithm using the metrics defined above. Table 2 shows the average of the $\boldsymbol{\epsilon}$-accuracy and $\boldsymbol{\epsilon}$-coverage ratios, calculated for different evaluation thresholds $\boldsymbol{\epsilon}'$. Using smaller $\boldsymbol{\epsilon}'$ values provides a stricter assessment.

Adaptive $\boldsymbol{\epsilon}$-PAL with its theoretical parameters ($h_{\max} = 24$) achieves high performance (99% average accuracy/coverage) when evaluated at its target $\boldsymbol{\epsilon}' = (0.05, 0.05)$. This aligns well with the theoretical

[3]`https://github.com/HIPS/Spearmint/tree/PESM`
[4]`https://github.com/belakaria/USeMO`

Table 2: Average of the $\boldsymbol{\epsilon}$-accuracy and coverage ratios (as percentages, mean ± 99%-confidence interval) for different evaluation thresholds $\boldsymbol{\epsilon}' = (\epsilon', \epsilon')$. Highest mean value in each row is bolded. Target accuracy for Adaptive $\boldsymbol{\epsilon}$-PAL was $\boldsymbol{\epsilon} = (0.05, 0.05)$.

$\boldsymbol{\epsilon}'$	**Adaptive $\boldsymbol{\epsilon}$-PAL** ($h_{\max} = 24$)	**Adaptive $\boldsymbol{\epsilon}$-PAL** ($h_{\max} = 10$)	**Adaptive $\boldsymbol{\epsilon}$-PAL** ($h_{\max} = 9$)	PESMO	ParEGO	USeMO
0.050	98 ± 2	$\mathbf{99 \pm 1}$	$\mathbf{99 \pm 1}$	92 ± 2	94 ± 1	92 ± 2
0.010	97 ± 2	$\mathbf{98 \pm 2}$	97 ± 1	90 ± 3	94 ± 1	74 ± 4
0.005	$\mathbf{97 \pm 1}$	$\mathbf{97 \pm 1}$	90 ± 3	90 ± 3	84 ± 4	60 ± 5
0.001	$\mathbf{78 \pm 4}$	64 ± 5	42 ± 5	54 ± 5	56 ± 5	26 ± 4

Table 3: Average mean-squared error (MSE) and total running time (hh:mm:ss) of the algorithms. Lowest MSE and runtime are bolded.

Metric	**Adaptive $\boldsymbol{\epsilon}$-PAL** ($h_{\max} = 24$)	**Adaptive $\boldsymbol{\epsilon}$-PAL** ($h_{\max} = 10$)	**Adaptive $\boldsymbol{\epsilon}$-PAL** ($h_{\max} = 9$)	PESMO	ParEGO	USeMO
MSE ($\times 10^{-6}$)	**5**	8	40	20	30	4500
Runtime	15:15:24	00:01:00	**00:00:27**	00:09:20	00:07:30	00:23:40

expectation of producing a valid $\boldsymbol{\epsilon}$-accurate Pareto set with high probability. The slight deviation from 100% can be attributed to the probabilistic nature of the guarantees ($\delta = 0.05$) and the approximations involved in discretizing the true Pareto front for metric calculation. Notably, reducing $h_{\max}$ to 10 or 9 maintains excellent performance at the target $\epsilon' = 0.05$. However, as expected, using a smaller $h_{\max}$ affects the performance at stricter evaluation thresholds ($\epsilon' < 0.05$), as the algorithm has less resolution to precisely delineate the Pareto front.

Comparing with competitors, Adaptive $\boldsymbol{\epsilon}$-PAL (even with $h_{\max} = 10$) achieves superior combined accuracy and coverage across the different evaluation thresholds. USeMO shows good accuracy for $\epsilon' = 0.05$ but degrades quickly and exhibits poor coverage, likely due to returning a sparse set of points. PESMO and ParEGO offer reasonable performance, with ParEGO slightly better overall in this setup, but neither consistently matches Adaptive $\boldsymbol{\epsilon}$-PAL's performance, especially at stricter thresholds.

Table 3 presents the average MSE and total running time. The MSE results corroborate the accuracy/coverage findings: Adaptive $\boldsymbol{\epsilon}$-PAL achieves the lowest MSE, indicating its predicted front is closest to the true front on average. The runs with $h_{\max} = 10$ and $h_{\max} = 24$ yield comparable MSE values, significantly better than competitors. USeMO has a notably high MSE, consistent with its poor coverage.

Observing the running times reveals the practical benefit of tuning $h_{\max}$. Reducing $h_{\max}$ from 24 to 10 decreased the runtime dramatically (by over 900 times in this instance) while preserving performance at the target $\epsilon = 0.05$. This demonstrates that Adaptive $\boldsymbol{\epsilon}$-PAL can be made computationally efficient by selecting a practical tree depth limit, while still offering a significant accuracy-advantage over the baseline methods tested.

8 Conclusion

In this paper, we proposed a new algorithm for PAL in large design spaces. Our algorithm learns an $\boldsymbol{\epsilon}$-accurate Pareto set of designs in as few evaluations as possible by combining an adaptive discretization strategy with GP inference of the objective values. We proved both information-type and metric-dimension type bounds on the sample complexity of our algorithm. To the best of our knowledge, this is the first sample complexity result for PAL that (i) involves an information gain term, which captures the dependence between objectives and (ii) explains how sample complexity depends on the metric dimension of the design space.

References

Shima Alizadeh, Aniruddha Bhargava, Karthick Gopalswamy, Lalit Jain, Branislav Kveton, and Ge Liu. Pessimistic off-policy multi-objective optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 2980–2988. PMLR, 2024.

Oscar Almer, Nigel Topham, and Björn Franke. A learning-based approach to the automated design of mpsoc networks. In *International Conference on Architecture of Computing Systems*, pp. 243–258. Springer, 2011.

Ryo Ariizumi, Matthew Tesch, Howie Choset, and Fumitoshi Matsuno. Expensive multiobjective optimization for robotics with consideration of heteroscedastic noise. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2230–2235. IEEE, 2014.

Peter Auer, Chao-Kai Chiang, Ronald Ortner, and Madalina Drugan. Pareto front identification from stochastic bandit feedback. In *Artificial intelligence and statistics*, pp. 939–947. PMLR, 2016.

Syrine Belakaria and Aryan Deshwal. Max-value entropy search for multi-objective bayesian optimization. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Syrine Belakaria, Aryan Deshwal, Nitthilan Kannappan Jayakodi, and Janardhan Rao Doppa. Uncertainty-aware search framework for multi-objective bayesian optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 10044–10052, 2020.

Syrine Belakaria, Alaleh Ahmadianshalchi, Barbara Engelhardt, Stefano Ermon, and Janardhan Rao Doppa. Non-myopic multi-objective bayesian optimization. *arXiv preprint arXiv:2412.08085*, 2024.

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *International conference on Algorithmic learning theory*, pp. 23–37. Springer, 2009.

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. X-armed bandits. *Journal of Machine Learning Research*, 12(May):1655–1695, 2011.

Róbert Busa-Fekete, Balázs Szörényi, Paul Weng, and Shie Mannor. Multi-objective bandits: Optimizing the generalized gini index. In *International Conference on Machine Learning*, pp. 625–634. PMLR, 2017.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. *arXiv preprint arXiv:2006.05078*, 2020.

Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *NIPS-Twenty-Sixth Annual Conference on Neural Information Processing Systems*, 2012.

Alkis Gotovos. Active learning for level set estimation. Master's thesis, Eidgenössische Technische Hochschule Zürich, Department of Computer Science,, 2013.

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*, pp. 1492–1501. PMLR, 2016.

C-L Hwang and Abu Syed Md Masud. *Multiple objective decision making—methods and applications: a state-of-the-art survey*, volume 164. Springer Science & Business Media, 2012.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

Julian Katz-Samuels and Clay Scott. Feasible arm identification. In *International Conference on Machine Learning*, pp. 2535–2543. PMLR, 2018.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 681–690, 2008.

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.

Cyrille Kone, Emilie Kaufmann, and Laura Richert. Adaptive algorithms for relaxed pareto set identification. *Advances in Neural Information Processing Systems*, 36:35190–35201, 2023.

H. T. Kung, F. Luccio, and F. P. Preparata. On finding the maxima of a set of vectors. *Journal of the ACM*, 22:469–476, 1975.

Xi Lin, Zhiyuan Yang, Xiaoyuan Zhang, and Qingfu Zhang. Pareto set learning for expensive multi-objective optimization. *Advances in Neural Information Processing Systems*, 35:19231–19247, 2022.

Andrea Locatelli, Maurilio Gutzeit, and Alexandra Carpentier. An optimal algorithm for the thresholding bandit problem. In *International Conference on Machine Learning*, pp. 1690–1698. PMLR, 2016.

Shie Mannor and John N Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun):623–648, 2004.

Subhojyoti Mukherjee, Anusha Lalitha, Sailik Sengupta, Aniket Deshmukh, and Branislav Kveton. Multi-objective alignment of large language models through hypervolume maximization. *arXiv preprint arXiv:2412.05469*, 2024.

Doruk Öner, Altuğ Karakurt, Atilla Eryılmaz, and Cem Tekin. Combinatorial multi-objective multi-armed bandit problem. *arXiv preprint arXiv:1803.04039*, 2018.

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

Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, and Markus Vincze. Multiobjective optimization on a limited budget of evaluations using model-assisted $\mathcal{S}$-metric selection. In *International Conference on Parallel Problem Solving from Nature*, pp. 784–794. Springer, 2008.

Ryan Roussel, Adi Hanuka, and Auralee Edelen. Multiobjective bayesian optimization for online accelerator tuning. *Physical Review Accelerators and Beams*, 24(6):062801, 2021.

R Scmidt. Dose-finding studies in clinical drug development. *European journal of clinical pharmacology*, 34 (1):15–19, 1988.

Amar Shah and Zoubin Ghahramani. Pareto frontier learning with expensive correlated objectives. In *International Conference on Machine Learning*, pp. 1919–1927. PMLR, 2016.

Shubhanshu Shekhar, Tara Javidi, et al. Gaussian process bandits with adaptive discretization. *Electronic Journal of Statistics*, 12(2):3829–3874, 2018.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

Ben Tu, Axel Gandy, Nikolas Kantas, and Behrang Shafei. Joint entropy search for multi-objective bayesian optimization. *Advances in Neural Information Processing Systems*, 35:9922–9938, 2022.

Mengfan Xu and Diego Klabjan. Pareto regret analyses in multi-objective multi-armed bandit. In *International Conference on Machine Learning*, pp. 38499–38517. PMLR, 2023.

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

Marcela Zuluaga, Andreas Krause, and Markus Püschel. ε-pal: an active learning approach to the multi-objective optimization problem. *The Journal of Machine Learning Research*, 17(1):3619–3650, 2016.