Learning Regions of Interest for Bayesian Optimization with Adaptive Level-Set Estimation

Fengxue Zhang[1] **Jialin Song**[2] **James Bowden**[3] **Alexander Ladd**[4] **Yisong Yue**[3] **Thomas A. Desautels**[4] **Yuxin Chen**[1]

Abstract

We study Bayesian optimization (BO) in high-dimensional and non-stationary scenarios. Existing algorithms for such scenarios typically require extensive hyperparameter tuning, which limits their practical effectiveness. We propose a framework, called BALLET, which adaptively filters for a high-confidence region of interest (ROI) as a superlevel-set of a nonparametric probabilistic model such as a Gaussian process (GP). Our approach is easy to tune, and is able to focus on local region of the optimization space that can be tackled by existing BO methods. The key idea is to use two probabilistic models: a coarse GP to identify the ROI, and a localized GP for optimization within the ROI. We show theoretically that BALLET can efficiently shrink the search space, and can exhibit a tighter regret bound than standard BO without ROI filtering. We demonstrate empirically the effectiveness of BALLET on both synthetic and real-world optimization tasks.

1. Introduction

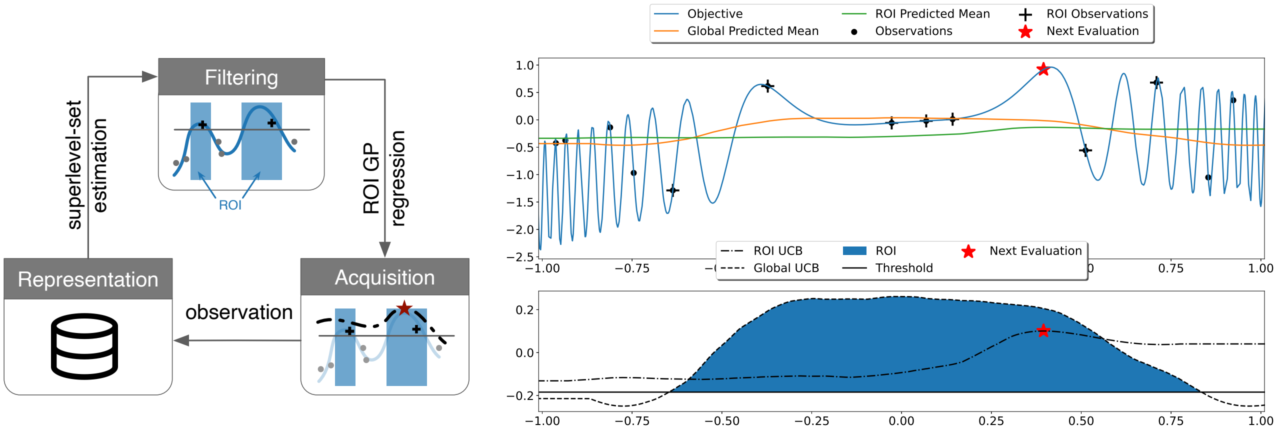
Bayesian optimization (BO) is a popular statistic-model-based sequential optimization method in various fields of science and engineering, including scientific experimental design (Yang et al., 2019), robotics planning (Berkenkamp et al., 2016; Sui et al., 2018), self-tuning systems (Zhang et al., 2022) and hyperparameter optimization (Snoek et al., 2012). These applications often involve optimizing a black-box function that is expensive to evaluate, where the statistics-guided efficient optimization algorithm is desired. The common practice in BO is to employ Gaussian processes (GPs) (Rasmussen & Williams, 2006) as a statistic surrogate model for the unknown objective function due to its mathematical simplicity as well as the promising capability in terms of learning and inference, which allows for defining effective acquisition functions.

Despite strong empirical and theoretical results under certain assumptions (e.g., smoothness) (Srinivas et al., 2009; Wang & Jegelka, 2017; Wang et al., 2016b), BO has struggled to achieve strong results in many real-world settings. First, the *high-dimensional*, *large-scale*, and *heterogeneous* characteristics of real-world optimization tasks remain key challenges for BO. Besides the well-known curse of dimensionality (Bengio et al., 2005), the heterogeneity and scarcity of training data in real-world tasks make it challenging to fit a single (global) GP for data acquisition (Eriksson et al., 2019). Meanwhile, purely relying on local characteristics has been proven to be ineffective for global optimization, due to the ignorance of the correlations on observations that are normally captured by the GP. The trade-off between exploiting data locality and exploring uncertainty at a global scale emerges as a critical problem in real-world BO settings, especially when the global smoothness assumption no longer holds.

Historically, various partitioning-based BO methods have been proposed to tackle this challenge. These methods, often based on certain clustering heuristics, learn the *regions of interest* (ROI) to better reflect the data locality. A common issue for existing heuristics is the added layer of complexity for model fine-tuning, which involves optimizing extra hyperparameters such as the number of ROIs (Eriksson et al., 2019), maximum leaf size in the tree-structured partitioning methods, and methods to generalize the partition learned on the accumulated observations to the whole search space (Munos, 2014).

We propose a novel nonparametric approach for partitioning-based BO that demonstrates strong empirical performance in real-world tasks, while having few hyperparameters to maintain. The proposed algorithm is inspired by the *level-set estimation* (LSE) problem, where a level-set corresponds to a set of points for which the black-box objective function takes value above (or below) some *given* threshold. Given a threshold, Gotovos et al. (2013) show that one can leverage

[1] Departmet of Computer Science, University of Chicago, Illinois, U.S. [2] Nvidia, California, U.S. [3] California Institute of Technology, California, U.S. [4] Lawrence Livermore National Laboratory, California, U.S.. Correspondence to: Yuxin Chen <chenyuxin@uchicago.edu>.



(a) The BALLET framework

(b) Illustration on a synthetic 1D function.

Figure 1: (a) Schematic of the algorithmic framework of BALLET. It first identifies the regions of interest by estimating the superlevel-set via a global GP. Then it trains a second GP on the superlevel-set ROIs and uses this GP to acquire the next data point to evaluate (marked by "★"). (b) illustrates a single iteration of BALLET. The upper figure shows the underlying objective function, together with the posterior mean for both the global GP and ROI GP. Training examples (i.e. observations) for the global GP and the ROI GP are marked by "●" and "✚" accordingly. The lower figure shows the filtering mechanism using the UCB (- - - -) of the global GP and the threshold determined by the maximal LCB (——) of the ROI GP (section 3.1, line 4 of algorithm 1). By learning the ROI GP on the filtered area, BALLET is guided by its acquisition function (—·—) to the ROIs rather than the "bad" regions which, with high confidence, is sub-optimal. The next data point is then chosen from the ROIs (★). Details for this 1D synthetic experiment are provided in section 4.

the point-wise confidence interval to actively identify the level-set with a theoretical guarantee. In the context of Bayesian optimization, the threshold could reduce to the lower confidence bound of the global optima.

Our contribution Following the above insight, we propose the novel Bayesian optimization framework with adaptive estimation of regions of interest. As illustrated in Figure 1, The algorithm partitions the search space based on confidence intervals and identifies the superlevel-set as the ROIs of high confidence contain the global optimum. We propose a novel acquisition function that relies on both the global model and the ROI model to capture the locality while not sacrificing global knowledge through optimization. We further provide rigorous theoretical analyses showing that the proposed acquisition function can, in a principled way, exhibit an improved regret bound compared to its canonical BO counterpart without the filtering component. We demonstrate the effectiveness of the proposed framework with an empirical study on several synthetic and real-world optimization tasks.

2. Related Work

High-dimensional Bayesian optimization BO often uses Gaussian processes as a (mathematically) simple yet powerful tool to parametrize the black-box function. However, GPs are difficult to fit in the high-dimensional setting due to the curse of dimensionality; thus classical BO algorithms need to be modified for high-dimensional function classes (Djolonga et al., 2013). A class of methods leverages additional structures, such as additive GPs, to mitigate the challenge of training a single global GP. For instance, LineBO restricts its search space to a one-dimensional subspace with reduced sample complexity at each step (Kirschner et al., 2019). GP-ThreDS relies on Hölder condition on the unknown objective to prune the search space, avoids the discretization cost exponential to the dimensionality of the search space, and speeds up the optimization (Salgia et al., 2021). In contrast, we aim at the applications where no Hölder smoothness is guaranteed, while valid discrete candidates in the search space are given. Another line of work is to embed the high-dimensional input space into a low-dimensional subspace (Wang et al., 2016a). Our proposed method relates most closely to methods with input space partitions (Wabersich & Toussaint, 2016; Eriksson et al., 2019; Wang et al., 2020; Sazanovich et al., 2021). Notably, TurBO (Eriksson et al., 2019) maintains a collection of local GPs and allocates queries with a multi-armed bandit procedure. LA-MCTS (Wang et al., 2020) learns a partition of the input space and uses Monte Carlo tree search (MCTS) to decide a subspace to apply BO. Compared with the proposed BALLET, these partitioning methods rely on heuristics and add extra complexity to the optimization task with hyperparameters of these heuristics, e.g., TuRBO relies on the number of trust regions and LA-MCTS relies on leaf size, a hyperparameter in UCB for the subspace selection and one for the partitioning algorithm.

Partition-based Bayesian active learning and optimization Partition-based methods are common in BO with safety constraints (Sazanovich et al., 2021; Sui et al., 2018; Makarova et al., 2021). These methods use LCB from GPs to partition the input space into safe and unsafe subspaces. Subsequent optimizaton queries are restricted to the safe subspaces only. Another related work is the level set estimation (LSE) method by Gotovos et al. (2013), where the authors use both UCB and LCB to narrow down regions where a particular function value is likely to exist. A unified framework for BO and LSE task (Bogunovic et al., 2016) proposes a similar filtering method but does not learn a local surrogate model. Instead, the filtering is used to constrain its acquisition function. Our method inherits the spirit of LSE to leverage the confidence interval to adaptively partition the search space.

Partition-based optimization methods More broadly speaking, partitioning the input space is a general strategy employed by several optimization methods (Munos, 2011; 2014; Shahriari et al., 2016; Merrill et al., 2021; Kawaguchi et al., 2016). Simultaneous optimistic optimization (SOO) algorithm (Munos, 2011; 2014), which is a non-Bayesian approach that intelligently partitions the space based on observed experiments to effectively balance exploration and exploitation of the objective. A modification of SOO, named Locally Oriented Global Optimization (LOGO) (Kawaguchi et al., 2016), achieves both fast convergence in practice and a finite-time error bound in theory. However, these non-Bayesian approaches have seen more degraded empirical performance on high-dimensional functions than their Bayesian counterparts (Merrill et al., 2021).

3. Bayesian Optimization with Adaptive Level-Set Estimation

We consider the standard BO setting for sequentially optimizing a function $f : \mathbf{X} \to \mathbb{R}$, where $\mathbf{X} \subseteq \mathbb{R}^d$ is the search space. At iteration t, we maintain a Gaussian process as the surrogate model, picks a point $\mathbf{x}_t \in \mathbf{X}$ by maximizing the acquisition function $\alpha : \mathbf{X} \to \mathbb{R}$, and observe the function value perturbed by additive noise: $y_t = f(\mathbf{x}_t) + \epsilon_t$ with $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ being i.i.d. Gaussian noise. The goal is to maximize the sum of rewards $\sum_{t=1}^{T} f(\mathbf{x}_t)$ over T iterations, or equivalently, to minimize the *cumulative regret* $R_T \triangleq \sum_{t=1}^{T} r_t$, where $r_t \triangleq \max_{\mathbf{x}\in\mathbf{X}} f(\mathbf{x}) - f(\mathbf{x}_t)$ denotes the *instantaneous regret*. Another common performance metric in BO is the *simple regret* $r_T^* \triangleq \max_{\mathbf{x}\in\mathbf{X}} f(\mathbf{x}) - \max_{t\leq T} f(\mathbf{x}_t)$.

3.1. The BALLET framework

Global modeling and representation Existing works use heuristics to partition the historical observations $\mathbf{D}_t = \{X_t, Y_t\}$ first and then generalize it to the whole search space $\mathbf{X}$ (Wang et al., 2020; Eriksson et al., 2019). Here $Y_t = \{y_1, ..., y_t\}$ and $X_t = \{\mathbf{x}_1, ..., \mathbf{x}_t\}$. The heuristics could be sensitive to additional hyperparameters of the partitioning model (e.g., number of partitions, etc), which in turn affect the optimization performance. Instead, we propose to learn a partitioning on $\mathbf{X}$ with a global estimation of the underlying blackbox function $f_{\text{g}} \triangleq f$, which is modeled by a Gaussian process $\mathcal{GP}_{f_{\text{g}}}(m_{f_{\text{g}}}(\mathbf{x}), k_{f_{\text{g}}}(\mathbf{x}, \mathbf{x}'))$ trained on the historical observations. $\mathcal{GP}_{f_{\text{g}}}$ is parameterized by $\theta_{f_{\text{g}}}$, where $m_{f_{\text{g}}}(\mathbf{x})$ is the mean function and $k_{f_{\text{g}}}(\mathbf{x}, \mathbf{x}')$ is the covariance function.

In this work, we resort to *Deep Kernel Learning* (DKL) (Wilson et al., 2016) as a scalable tool to train the GPs. [1] The algorithm learns a latent space mapping $q : \mathbf{X} \to \mathbf{Z}$ on a neural network to convert the input space $\mathbf{X}$ to the latent space $\mathbf{Z}$, and constructs an objective mapping $h : \mathbf{Z} \to \mathbb{R}$ such that $f(\mathbf{x}) \approx h(q(\mathbf{x}))$, $\forall \mathbf{x} \in \mathbf{X}$. The neural network q and the base kernel k together are regarded as a *deep kernel*, denoted by $k_{f_{\text{g}}}(\mathbf{x}, \mathbf{x}') = k\left(q(\mathbf{x}), q(\mathbf{x}')\right)$ (Wilson et al., 2016). The deep kernel is trained by maximizing the negative log-likelihood (NLL) $-\log(\mathbb{P}\left[\mathbf{y}_t \mid X_t, \theta_{f_{\text{g}},t}\right]) = -\frac{1}{2}\mathbf{y}_t^\top(\mathbf{K}_{f_{\text{g}},t} + \sigma^2 I)^{-1}\mathbf{y}_t - \frac{1}{2}\log|(\mathbf{K}_{f_{\text{g}},t} + \sigma^2 I)| - \frac{t}{2}\log(t)$ which is the learning objective for the kernel (Rasmussen & Williams, 2006). In addition to DKL, we use the unlabeled dataset which is sampled from $\mathbf{X}$ to pre-train an Auto-Encoder and use the parameters of its encoder to initialize the neural network q following the protocol described by Ferreira et al. (2020).

At iteration t, given the selected points $\mathbf{D}_t$, the posterior over f_{g} also takes the form of a GP, with mean $\mu_{f_{\text{g}},t}(\mathbf{x}) = k_{f_{\text{g}},t}(\mathbf{x})^\top(\mathbf{K}_{f_{\text{g}},t} + \sigma^2 I)^{-1}\mathbf{y}_t$ and covariance $k_{f_{\text{g}},t}(\mathbf{x}, \mathbf{x}') = k_{f_{\text{g}}}(\mathbf{x}, \mathbf{x}') - k_{f_{\text{g}},t}(\mathbf{x})^\top(\mathbf{K}_{f_{\text{g}},t} + \sigma^2 I)^{-1}k_{f_{\text{g}},t}(\mathbf{x}')$, where $k_{f_{\text{g}},t}(\mathbf{x}) \triangleq \left[k_{f_{\text{g}}}(\mathbf{x}_1, \mathbf{x}), \ldots, k_{f_{\text{g}}}(\mathbf{x}_t, \mathbf{x})\right]^\top$ and $\mathbf{K}_{f_{\text{g}},t} \triangleq \left[k_{f_{\text{g}}}(\mathbf{x}, \mathbf{x}')\right]_{\mathbf{x},\mathbf{x}'\in\mathbf{D}_t}$ is the positive definite kernel matrix (Rasmussen & Williams, 2006).

Superlevel-set estimation and filtering The global $\mathcal{GP}_{f_{\text{g}}}$ induces a filter on $\mathbf{X}$ to locate the region of interest $\hat{\mathbf{X}}$. It is desired for $\hat{\mathbf{X}}$ that with high probability, the optimum $\mathbf{x}^* \in \arg\max_{\mathbf{x}\in\mathbf{X}} f(\mathbf{x})$ is contained in $\hat{\mathbf{X}}$, while $\|\hat{\mathbf{X}}\| \ll \|\mathbf{X}\|$. Specifically, we leverage the confidence interval of the global Gaussian process $\mathcal{GP}_{f_{\text{g}}}$ to define the upper confidence bound $\text{UCB}_{f_{\text{g}},t}(\mathbf{x}) \triangleq \mu_{f_{\text{g}},t-1}(\mathbf{x}) + \beta_t^{1/2}\sigma_{f_{\text{g}},t-1}(\mathbf{x})$ and lower confidence bound $\text{LCB}_{f_{\text{g}},t}(\mathbf{x}) \triangleq \mu_{f_{\text{g}},t-1}(\mathbf{x}) - \beta_t^{1/2}\sigma_{f_{\text{g}},t-1}(\mathbf{x})$, where $\sigma_{f_{\text{g}},t-1}(\mathbf{x}) = k_{f_{\text{g}},t-1}(\mathbf{x}, \mathbf{x})^{1/2}$

[1] We propose a kernel-agnostic framework and in implementation, we apply the efficient deep kernel for large-scale optimization. In deep kernel learning, which is shown to bear strong empirical performance in regression and optimization (e.g.(Wilson et al., 2016; Wistuba & Grabocka, 2021)), the learning cost is $\mathcal{O}(n)$ for n training points, and the prediction cost is $\mathcal{O}(1)$ per test point and is more efficient than exact GP in terms of computational cost.

Algorithm 1 Bayesian Optimization with Adaptive Level-Set Estimation (BALLET)

```
Input:Search space $\mathbf{X}$, initial observation $\mathbf{D}_0$, horizon $T$;
for $t = 1$ to $T$ do
    Fit the global Gaussian process $\mathcal{GP}_{f_{\mathrm{g}},t}$: $\theta_{f_{\mathrm{g}},t} \leftarrow \arg\max_{\theta_{f_{\mathrm{g}}}} -\log \mathbb{P}\left[Y_t \mid X_{t-1}, \theta_{f_{\mathrm{g}}}\right]$
    Identify ROIs via superlevel-set estimation $\hat{\mathbf{X}}_t \leftarrow \{\mathbf{x} \in \mathbf{X} \mid \mathrm{UCB}_{f_{\mathrm{g}},t}(\mathbf{x}) \geq \mathrm{LCB}_{f_{\mathrm{g}},t,\max}\}$
    Partition the historical observation: $\hat{\mathbf{D}}_t \leftarrow \{(\mathbf{x}, y) \in \mathbf{D}_t \mid \mathbf{x} \in \hat{\mathbf{X}}_t\}$.
    Fit the ROI Gaussian process $\mathcal{GP}_{\hat{f},t}$: $\theta_{\hat{f},t} \leftarrow \arg\max_{\theta_{\hat{f}}} -\log \mathbb{P}\left[Y_t \cap \hat{\mathbf{D}}_t | X_t \cap \hat{\mathbf{D}}_t, \theta_{\hat{f}}\right]$
    Optimize the superlevel-set acquisition function: $\mathbf{x}_{t+1} \leftarrow \arg\max_{\mathbf{x} \in \hat{\mathbf{X}}} \alpha_{\hat{f}}(\mathbf{x})$ (e.g., as defined in equation 7, 8 or 9)
    $\mathbf{D}_{t+1} \leftarrow \mathbf{D}_t \cup \{(\mathbf{x}_{t+1}, y_{t+1})\}$
end for
Output: $\max_t y_t$
```

and β acts as an scaling factor. Then the maximum of the global lower confidence bound $\mathrm{LCB}_{f_{\mathrm{g}},t,\max} \triangleq \max_{\mathbf{x}\in\mathbf{X}} \mathrm{LCB}_{f_{\mathrm{g}},t}(\mathbf{x})$ can be used as the threshold, and we attain the superlevel-set

$$\hat{\mathbf{X}}_t \triangleq \left\{\mathbf{x} \in \mathbf{X} \mid \mathrm{UCB}_{f_{\mathrm{g}},t}(\mathbf{x}) \geq \mathrm{LCB}_{f_{\mathrm{g}},t,\max}\right\} \tag{1}$$

as the region(s) of interest. The historical observation on this subset is denoted as

$$\hat{\mathbf{D}}_t \triangleq \left\{(\mathbf{x}, y) \in \mathbf{D} \mid \mathbf{x} \in \hat{\mathbf{X}}_t\right\}. \tag{2}$$

Remark 1 BALLET *is not assuming that the resulting* $\hat{\mathbf{X}}$ *is composed of one single cluster.* BALLET *learns a single GP over* $\hat{\mathbf{X}}_t$ *and optimizes on all these localities at the same time. Intuitively it aims at conducting (local) BO on the top tier (which could consist of multiple regions) of the unknown function. This mechanism avoids being overconfident to identify only one region of interest or the need to manually specify the number of clusters beforehand. Here we use the term "superlevel-set" to differentiate from the methods conducting local BO.*

3.2. BALLET-ICI

The goal of the filtering step in BALLET is to shrink the search space $\hat{\mathbf{X}}_t$ while ensuring that the optimum is contained in the ROIs with high probability. By definition of $\hat{\mathbf{X}}_t$ (equation 1), we note that at iteration t the size of the search space $\|\hat{\mathbf{X}}_t\|$ is directly affected by $\mathrm{UCB}_{f_{\mathrm{g}},t}(\mathbf{x})$. We thus consider the width of the range of $\mathrm{UCB}_{f_{\mathrm{g}},t}$ over $\mathbf{x} \in \hat{\mathbf{X}}_t$, formally defined as

$$\Delta_{\mathrm{ROI},t} \triangleq \max_{\mathbf{x}\in\hat{\mathbf{X}}} \mathrm{UCB}_{f_{\mathrm{g}},t}(\mathbf{x}) - \mathrm{LCB}_{f_{\mathrm{g}},t,\max} \tag{3}$$

as a surrogate objective to minimize.

3.2.1. Acquisition function

Evaluating equation 3 for a new data point $\mathbf{x}$ requires 1-step look-ahead (i.e., computing the expected $\Delta_{\mathrm{ROI},t+1}$ should $\mathbf{x}$ be acquired at t), which could be expensive. Instead, we consider the *point-wise confidence interval* of the ROI Gaussian process $\mathcal{GP}_{\hat{f}}$ trained on $\hat{\mathbf{D}}$, denoted by

$$\mathrm{CI}_t(\mathbf{x}) \triangleq \left[\mathrm{LCB}_{\hat{f},t}(\mathbf{x}), \mathrm{UCB}_{\hat{f},t}(\mathbf{x})\right], \tag{4}$$

and simply use the width of $|\mathrm{CI}_t(\mathbf{x})|$ as an efficiently proxy for evaluating $\mathbf{x}$.

Mitigating the loss of information of $\mathcal{GP}_{\hat{f}}$ At each iteration, BALLET conducts superlevel-set estimation and then runs BO on $\hat{\mathbf{X}}$ using the ROI Gaussian process $\mathcal{GP}_{\hat{f}}$ trained on $\hat{\mathbf{D}}$. Note that $\mathcal{GP}_{\hat{f}}$ could better capture the locality at the cost of *losing partial historical observations* due to the filtering as $\hat{\mathbf{D}} \subseteq \mathbf{D}$. The missing historical observations could result in additional undesired uncertainty in $\mathcal{GP}_{\hat{f}}$ compared to the global GP $\mathcal{GP}_{f_{\mathrm{g}}}$. To avoid such information loss while taking the advantage of the identified ROIs, we propose to tighten the confidence interval (equation 4) by taking the intersection of the confidence intervals from all ROI GPs trained from each of the previous iterations $\mathcal{GP}_{\hat{f},i\leq t}$ and the corresponding global GPs, $\mathcal{GP}_{f_{\mathrm{g}},i\leq t}$. In this way, the acting superlevel-set confidence interval would be $\widehat{\mathrm{CI}}_t(\mathbf{x}) \triangleq \left[\widehat{\mathrm{LCB}}_t(\mathbf{x}), \widehat{\mathrm{UCB}}_t(\mathbf{x})\right]$, where

$$\begin{cases} \widehat{\mathrm{LCB}}_t(\mathbf{x}) \triangleq \max_{i\leq t, f\in\{\hat{f},f_{\mathrm{g}}\}} \mathrm{LCB}_{f,i}(\mathbf{x}) \\ \widehat{\mathrm{UCB}}_t(\mathbf{x}) \triangleq \min_{i\leq t, f\in\{\hat{f},f_{\mathrm{g}}\}} \mathrm{UCB}_{f,i}(\mathbf{x}) \end{cases} \tag{5}$$

It is possible that the intersection in equation 5 results in an empty CI due to the dynamics brought by the learned kernels. In practice, instead of taking the intersections of all the historical GPs, we could mitigate the problem by only taking the intersection of the CIs at step t to get $\widetilde{\mathrm{CI}}_t(\mathbf{x}) = [\widetilde{\mathrm{LCB}}_t(\mathbf{x}), \widetilde{\mathrm{UCB}}_t(\mathbf{x})]$. Here

$$\begin{cases} \widetilde{\mathrm{LCB}}_t(\mathbf{x}) \triangleq \max_{f\in\{\hat{f},f_{\mathrm{g}}\}} \mathrm{LCB}_{f,t}(\mathbf{x}) \\ \widetilde{\mathrm{UCB}}_t(\mathbf{x}) \triangleq \min_{f\in\{\hat{f},f_{\mathrm{g}}\}} \mathrm{UCB}_{f,t}(\mathbf{x}) \end{cases} \tag{6}$$

Note when $\mathrm{LCB}_{\hat{f},t} \leq \mathrm{LCB}_{f_{\mathrm{g}},t}$ and LCB is monotonically increasing wrt t, it holds that $\mathrm{LCB}_{f_{\mathrm{g}},t,\max} = \max_{\mathbf{x}\in\hat{\mathbf{X}}} \widehat{\mathrm{LCB}}_t(\mathbf{x}) = \max_{\mathbf{x}\in\hat{\mathbf{X}}} \widetilde{\mathrm{LCB}}_t(\mathbf{x})$.

The BALLET-ICI acquisition function We propose to apply the intersection of the confidence intervals as an acquisition function for BALLET (BALLET-ICI), namely

$$\alpha_{\hat{f}}(\mathbf{x}) \triangleq \widetilde{\text{UCB}}_t(\mathbf{x}) - \widetilde{\text{LCB}}_t(\mathbf{x}) \tag{7}$$

Our algorithm is presented in Algorithm 1. In the following subsection, we rigorously justify the use of equation 7 as our acquisition function, and prove that the cost on the optimization performance using the relaxation from equation 5 to equation 6 could be bounded under certain conditions.

3.2.2. Theoretical analysis

By abuse of notation, we let the maximum confidence interval on a certain set denoted by

$$\widehat{\text{CI}}_{t,\max}(\cdot) = \left[\max_{\mathbf{x}\in\cdot} \widehat{\text{LCB}}_t(\mathbf{x}), \max_{\mathbf{x}\in\cdot} \widehat{\text{UCB}}_t(\mathbf{x})\right]$$

$$\widetilde{\text{CI}}_{t,\max}(\cdot) = \left[\max_{\mathbf{x}\in\cdot} \widetilde{\text{LCB}}_t(\mathbf{x}), \max_{\mathbf{x}\in\cdot} \widetilde{\text{UCB}}_t(\mathbf{x})\right]$$

The following lemma shows that the interval $\widehat{\text{CI}}_{t,\max}(\hat{\mathbf{X}})$ is a high confidence interval for $f^* = \max\limits_{\mathbf{x}\in\mathbf{X}} f(\mathbf{x})$ given a good discretization of the search space.

Lemma 1 *Assume $\forall t < T, \mathbf{x} \in \mathbf{X}$, $f(\mathbf{x})$ is a sample from global $\mathcal{GP}_{f_g,t}$. For any $\delta \in (0,1)$ and any finite discretization $\tilde{D}$ of $\mathbf{X}$ containing the optimum $\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathbf{X}} f(\mathbf{x})$, with $\beta_t = 2\log(2|\tilde{D}|\pi_t/\delta)$ where $\sum_{t\geq 1}\pi_t^{-1} = 1$, $\mathbb{P}\left[f^* \in \widehat{CI}_{t,\max}(\tilde{D})\right] \geq 1-\delta$.*

A proper choice of π_t satisfying Lemma 1 is $\pi_t = \frac{\pi^2 t^2}{6}$. The following corollary shows that with high probability, the global optimum is contained in the interval.

Remark 2 *Rigorously, $\forall t < T, \mathbf{x} \in \hat{\mathbf{X}}$, the marginalized $\mathcal{GP}_{\hat{f},t}$ and $\mathcal{GP}_{f_g,t}$ shall be the same. Therefore, $\forall t < T, \mathbf{x} \in \hat{\mathbf{X}}$, $f(\mathbf{x})$ is a sample from $\mathcal{GP}_{\hat{f},t}$ as well. However, in practice, it is challenging to specify the ideal prior. We introduce $\mathcal{GP}_{\hat{f},t}$ into the analysis to reflect the benefits of learning the hyperparameters for each GP separately in real-world scenarios.*

Corollary 1 *With the same conditions as in Lemma 1, $\mathbb{P}\left[\mathbf{x}^* \in \hat{\mathbf{X}}_t\right] \geq 1-\delta, \forall t \geq 1$.*

For simplification, we use the notation $\tilde{D}_{\hat{\mathbf{X}}} = \tilde{D} \cap \hat{\mathbf{X}}$. Taking the union bound over Lemma 1 and Corollary 1, we obtain the following result:

Corollary 2 *With probability at least $1-2\delta$, the global optimum lies in the following interval $\widehat{CI}_{t,\max}(\tilde{D}_{\hat{\mathbf{X}}}) \subseteq \widetilde{CI}_{t,\max}(\tilde{D}_{\hat{\mathbf{X}}})$.*

Corollary 2 indicates that by narrowing the interval, we could achieve efficient filtering in BALLET and identify the near-optimal areas. Define the maximum information gain about unknown function f after T rounds as $\gamma_{f,T} = \max_{A\subset\tilde{D}:|A|=T} \mathbb{I}(y_A; f_A)$. Also, define

$$\widehat{\gamma_T} = \min_{f\in\{f_g,\hat{f}\}} \gamma_{f,T}$$

The following results shows that $|\widehat{\text{CI}}_t(\mathbf{x})| = \widehat{\text{UCB}}_t(\mathbf{x}) - \widehat{\text{LCB}}_t(\mathbf{x})$ serves the purpose of efficiently narrowing the interval and the resulting range of it is bounded by $\widehat{\gamma}_T$.

Proposition 1 *Under the same conditions assumed in Lemma 1 except for $\beta_t = 2\log(2|\tilde{D}_{\hat{\mathbf{X}}}|\pi_t/\delta)$, with acquisition function $|\widehat{CI}_t(\mathbf{x})| = \widehat{UCB}_t(\mathbf{x}) - \widehat{LCB}_t(\mathbf{x})$, after at most $T \geq \frac{\beta_T\widehat{\gamma}_T C_1}{\epsilon^2}$ iterations, $\mathbb{P}\left[|\widehat{CI}_{T,\max}(\tilde{D}_{\hat{\mathbf{X}}})| \leq \epsilon\right] \geq 1-2\delta$. Here $C_1 = 8/\log(1+\sigma^{-2})$.*

The proposition reveals two potential improvements over the global GP-UCB (Srinivas et al., 2009) on the regret bound brought by BALLET. First, β_T takes smaller value due to the filtering compared to *Lemma* 1 which is also the term in the regret bounds of (Srinivas et al., 2009). Second, $\widehat{\gamma_T}$ could potentially be smaller than the global $\gamma_{f_g,T}$ with proper kernel learning on ROI. The following corollary shows the cost of using $\widetilde{\text{UCB}}_t(\mathbf{x}) - \widetilde{\text{LCB}}_t(\mathbf{x})$ as acquisition function is C_2^2 compared to $\widehat{\text{UCB}}_t(\mathbf{x}) - \widehat{\text{LCB}}_t(\mathbf{x})$.

Corollary 3 *Under the same conditions assumed in Lemma 1 except for $\beta_t = 2\log(2|\tilde{D}_{\hat{\mathbf{X}}}|\pi_t/\delta)$, with acquisition function $\alpha_{\hat{f}}(\mathbf{x}) = |\widetilde{CI}_t(x)| = \widetilde{UCB}_t(\mathbf{x}) - \widetilde{LCB}_t(\mathbf{x})$, after at most $T \geq \frac{\beta_T\widehat{\gamma}_T C_1 C_2^2}{\epsilon^2}$ iterations, $\mathbb{P}\left[|\widetilde{CI}_{T,\max}(\tilde{D}_{\hat{\mathbf{X}}})| \leq \epsilon\right] \geq 1-2\delta$. Here $C_1 = 8/\log(1+\sigma^{-2})$, and $C_2 = \frac{\min_{t\leq T}(|\widetilde{CI}_{t,\max}(\tilde{D}_{\hat{\mathbf{X}}})|)}{|\widetilde{CI}_{T,\max}(\tilde{D}_{\hat{\mathbf{X}}})|}$.*

The proof of the corollary follows exactly the same steps as the proof of Proposition 1 except leveraging the fact $\min_{t\leq T}(|\widetilde{\text{CI}}_{t,\max}(\tilde{D}_{\hat{\mathbf{X}}})|) = C_2|\widetilde{\text{CI}}_{T,\max}(\tilde{D}_{\hat{\mathbf{X}}})|$ at its last step.

3.3. Other BALLET variants

BALLET provides a flexible framework for partitioning-based BO. In addition to BALLET-ICI, one can run Thompson sampling on ROI as the acquisition function (BALLET-RTS), namely

$$\alpha_{\hat{f},\text{TS}}(\mathbf{x}) \triangleq \hat{f}_t(\mathbf{x}) \tag{8}$$

where $\hat{f}_t \sim GP_{\hat{f},t}$. Another BALLET variant is to directly run uncertainty sampling with $\mathcal{GP}_{\hat{f}}$ on $\hat{\mathbf{X}}$ (BALLET-RCI, where RCI is short for "ROI-CI"):

$$\alpha_{\hat{f},\text{RCI}}(\mathbf{x}) \triangleq |\text{CI}_t(\mathbf{x})| = \text{UCB}_{\hat{f},t}(\mathbf{x}) - \text{LCB}_{\hat{f},t}(\mathbf{x}) \tag{9}$$

Compared with BALLET-RTS and BALLET-RCI, the intersection of CIs defined in equation 7 in BALLET-ICI leverages the posterior information of both $\mathcal{GP}_{\hat{f}}$ and $\mathcal{GP}_{f_g}$. This allows BALLET-ICI to efficiently narrow the confidence interval for f^* by explicitly balancing exploration and exploitation, and achieve a high-probability theoretical guarantee on its optimization performance. We also discuss taking the UCB of the intersection of CI's as the acquisition function (BALLET-IUCB) in the appendix.

4. Experiment

Experimental setup We compare three baseline algorithms in our experiments against BALLET-ICI, BALLET-RCI, and BALLET-RTS. The Deep-Kernel-based Bayesian Optimization initialized with a pre-trained AutoEncoder (DKBO-AE) applies the deep kernel where a pre-trained AutoEncoder [2] initializes the neural network q (Zhang et al., 2022). The neural network consists of three hidden layers with 1000, 500, and 50 neurons, and ReLU non-linearity respectively. The output layer is one-dimensional. We use squared exponential kernel or linear kernel as the base kernel, i.e. $k_{\text{SE}}(\mathbf{x}, \mathbf{x}') = \sigma_{\text{SE}}^2 \exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{2l})$ or $k_{\text{LINEAR}}(\mathbf{x}, \mathbf{x}') = \sigma_{\text{LINEAR}}^2(\mathbf{x}^T\mathbf{x})$, for the deep kernel, and Thompson Sampling (Chapelle & Li, 2011) for the acquisition function α. Two other partition-based BO algorithms LA-MCTS (Wang et al., 2020) and TuRBO (Eriksson et al., 2019) serve as the baselines. Note that DKBO-AE is used as the subroutine for LA-MCTS, TuRBO-DK, and BALLET-RTS. The neural network architecture, base kernel and acquisition function are the same. BALLET-RCI and BALLET-ICI also share the same deep kernel except for applying different acquisition functions. The comparison between BALLET-RTS and DKBO-AE serves as the ablation study of **the proposed partitioning method**. The comparison between BALLET-RCI and BALLET-ICI also serves as the ablation study of **taking the intersection of CI** as defined in equation 6.

One crucial problem in practice is tuning the hyperparameters. For each of the algorithms, the same 10 randomly picked points serve as the warm-up set. We take the default hyperparameters from the open-sourced LA-MCTS [3] and TuRBO [4] implementation. Note that we choose TuRBO-1 implementation for TuRBO where there is one trust region through the optimization, as previous work has shown its robust performance in various tasks (Eriksson et al., 2019). For BALLET-ICI, we set δ in Lemma 1 to be 0.2. In addition, we find that using the $\beta_t^{1/2}$ in Lemma 1 to identify the ROI could be over-conservative in that it can not filter many areas and let BALLET-ICI regress to DKBO-AE with two similar GPs. Through the experiments, we fix $\beta_t^{1/2} = 0.2$ only when identifying ROIs as in line 4 of Algorithm 1. For all the tested algorithms, the base kernels are squared exponential kernels except for Nanophotonics and Water Converter where we applied linear kernels as the base kernel. We defer the detailed study of parameter choices in BALLET-ICI to the appendix.

Datasets *1D-Toy.* We create a synthetic dataset 1D-Toy of one dimension to illustrate the process of BALLET-ICI as is shown in section 3. The function is defined on $\mathbf{x} \in [-1, 1]$ as $f(\mathbf{x}) = \sin(64|\mathbf{x}|^4) - (\mathbf{x} - 0.2)^2$. This toy function consists of two high-frequency areas on both sides and a low-frequency area in the middle. The neural network is pre-trained on 100 data points.

HDBO-200D. We create a synthetic dataset Sum-200D of 200 dimensions. Each dimension is independently sampled from a standard normal distribution to maximize the uncertainty on that dimension and examine the algorithm's capability to solve the medium-dimensional problem. We want to maximize the label $f(\mathbf{x}) = \sum_{i=1}^{200} e^{x_i}$ which bears an additive structure and of non-linearity. The neural network is pre-trained on 100 data points.

Water Converter Configuration-16D. This UCI dataset we use consists of positions and absorbed power outputs of wave energy converters (WECs) from the southern coast of Sydney. The applied converter model is a fully submerged three-tether converter called CETO. 16 WECs locations are placed and optimized in a size-constrained environment. Note its values are at the order of $O(10^6)$.

Nanophotonics Structure Design. We wish to optimize a weighted figure of merit quantifying the fitness of the transmission spectrum for hyperspectral imaging as assessed by a numerical solver (Song et al., 2018). This problem has a 5-dimensional input corresponding to the physical design dimensions of a potential filter. Although the input is not high-dimensional, the function represents a discrete solution of Maxwell's equations and has a complex value landscape.

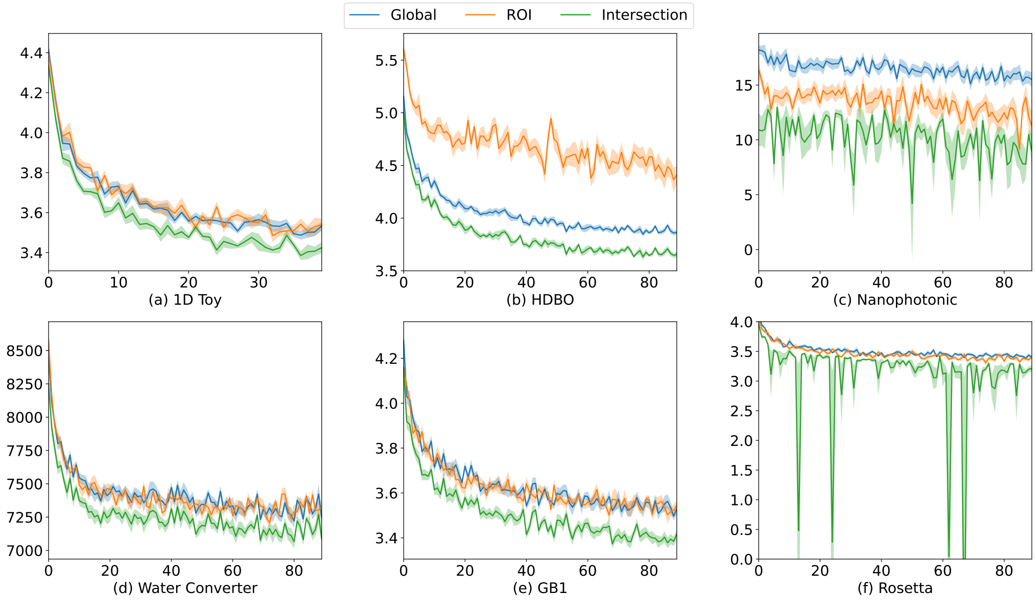
GB1. We use one protein dataset in which the objective is to maximize stability fitness predictions for the Guanine nucleotide-binding protein GB1 given different sequence mutations in a target region of 4 residues (Wu et al., 2019). Specifically, we use the ESM embedding generated by a transformer protein language model (Rives et al., 2021).

Rosetta Protein Design. We use another protein engineering dataset describing a set of antigen/antibody binding calculations. These calculations, executed using supercomputing resources, estimate the change in binding free energy at the interface between 71769 modified antibodies and the SARS-CoV-2 spike protein, as compared to a reference antibody. Estimations of binding free energy ($\Delta\Delta G$) are calculated using protein-structure-based Rosetta Flex simulation software (Das & Baker, 2008; Barlow et al., 2018). These

[2] The AutoEncoder is trained with random unlabelled samples.

[3] https://github.com/facebookresearch/LaMCTS

[4] https://botorch.org/tutorials/turbo_1



(a) 1D Toy

(b) HDBO

(c) Nanophotonic

(d) Water Converter

(e) GB1

(f) Rosetta

Figure 2: The confidence interval of f^* defined in Corollary 2 on each task is shown here. The results from each task are collected from at least 10 independent trials. The error bar demonstrates the standard error. As β varies on different iteration and different search space size, we fix $\beta_t = 2$ for comparable illustration. The x-axis denotes the number of iterations, and the y-axis denotes the width of the confidence interval.

calculations take several CPU hours each and are produced during an antibody design process (Desautels et al., 2020). Inputs are described with an 80-dimensional feature vector that, relative to the reference sequence, describes changes in the interface between the antibody and the corresponding target region on the SARS-CoV-2 spike. This is a particularly relevant problem setting when trying to rapidly choose antibody candidates to respond to a new disease in a timely fashion.

Confidence Intervals As is shown in figure 2, the CIs of f^* through the optimization of BALLET-ICI do not constantly narrow. Instead, on Nanophotonics the width generally remains the same through the 90 iterations, indicating the challenge of fitting these datasets with limited data points and therefore optimizing it with the underfitted GPs.

The intersection CI is consistently narrower than the other two, where there is no consistent superiority against each other. Though on HDBO the ROI curve is above the global curve, the resulting intersection CI still improves upon global CI, indicating that the maximizer of $\widetilde{\text{UCB}}_t$ and $\widetilde{\text{LCB}}_t$ are different for global GP and ROI GP.

The dynamics of kernel learning results in empty intersections on Rosetta, where occasionally the width of CI for f^* turns out to be zero, showing the potential problem in taking the intersection of all historical CIs. Future improvement on BALLET-ICI could be better aligning the CI of both ROI and global GPs through different iterations to allow taking the intersection of all historical CIs as in Proposition 1.

The intersection curve converges faster to non-zero values on both HDBO and GB1 showing the benefits of taking the intersection of global CI and ROI CI as it better captures the localities with ROI GP while not losing information of global GP. However, the width of CI could not directly serve as the indicator for the optimization performance. On GB1, the intersection curve is uniformly better than both the ROI and global CI, while in figure 3, BALLET-ICI doesn't outperform DKBO-AE, BALLET-RTS, or BALLET-ROI-UCB as the CI for f^* is still larger than 3.3.

Optimization Performance The experiment results in figure 3 demonstrate the robust performance of BALLET-ICI which consistently matches or outperforms the best baseline. In contrast, LA-MCTS consistently matches or outperforms TuRBO-DK, but lags behind DKBO-AE on the 1D Toy which indicates its potential inefficiency in the tasks of high-frequency areas hindering its partitioning of the search space. Note that we also find that using SVM to generalize the partition on Y_t to $\mathbf{X}$ in LA-MCTS occasionally fails possibly due to the intrinsic complexity of the partition learned on Y_t demanding methods of greater capability, while the level-set partition of BALLET-ICI is regularized by the smoothness of the global Gaussian process $\mathcal{GP}_{f_g}$. We reject the failed LA-MCTS trials.

TuRBO-DK matches BALLET-ICI performance on GB1 and but loses to DKBO-AE on all other cases. By construction, 1D Toy and HDBO-200D could have a large amount of distant local maximum, while TuRBO-DK relies on the locality of the observation to identify the trust regions. TuRBO-DK could be potentially trapped in the local maximum and the performance degrades in the scenario where the multiple modules are distant from each other while the

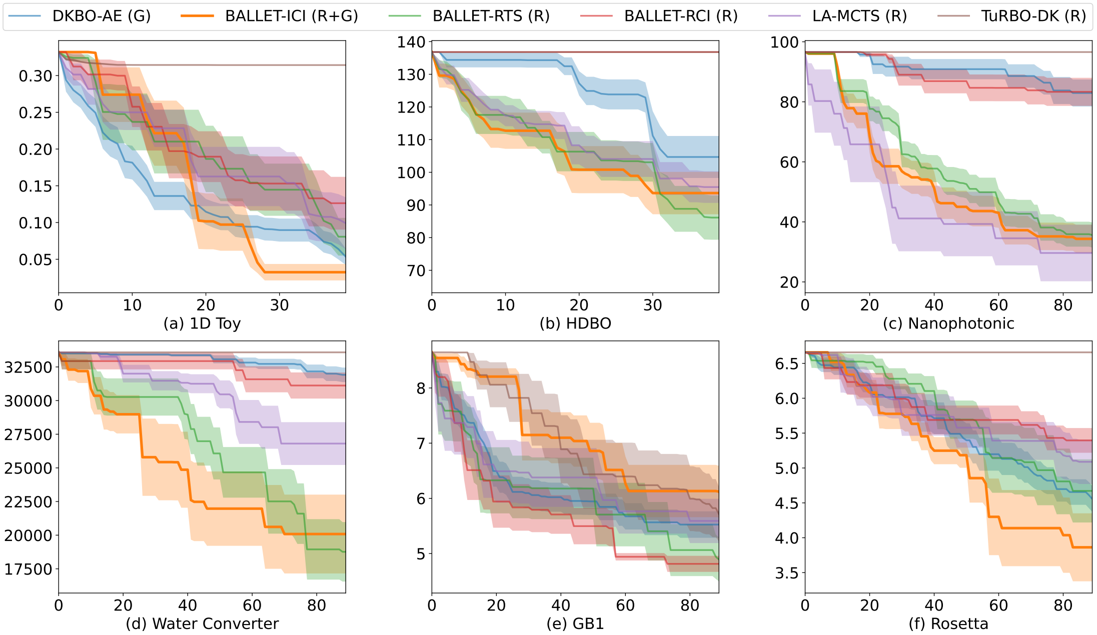


Figure 3: Simulation results on each task are shown here. The results from each task are collected from at least 10 independent trials. The error bar demonstrates the standard error. The x-axis denotes the number of iterations, and the y-axis denotes the simple regret. The simple regrets for the 10 initial randomly picked warm-up datasets are clipped. (G), (R), and (R+G) means the global model only, the ROI model only, and the ROI model combined with the global model correspondingly.

gap between sub-optimal and optimal observation is significant. In contrast, BALLET-ICI is capable of identifying multiple regions of interest with the level-set partitioning without specifying the desired number of regions.

On 1D-Toy dataset which is composed of the low-value high-frequency areas and the high-value low-frequency area, BALLET-ICI significantly outperforms the baselines and reaches the near-optimal area within 30 iterations. Due to the complexity of the low-value areas that make up a large portion of the objective, the GPs suffer from the under-fitting, especially at the beginning stage where access to observation is limited as is shown in figure 1. At this phase, the DKBO-AE stably outperforms BALLET-ICI potentially without the distraction from the under-fitting ROI GP. While on HDBO datasets which by construction bears relative uniform smoothness, the partition-based algorithms other than TuRBO-DK all enjoy similar benefits at the beginning stage.

On Nanophotonics, Water-Converter, and Rosetta, BALLET-ICI matches or outperforms the baselines including BALLET-RCI, while losing to BALLET-RCI and BALLET-RTS on GB1. This ablation study indicates the necessity of taking the intersection of CI in most scenarios, while also revealing that more aggressive filtering of BALLET-RCI could sometime be beneficial. BALLET-RTS matches BALLET on HDBO, Nanophotonics, Water-Converter, and Rosetta reflects that the ROI GP could be equivalently informative as the combination of both the global GP and ROI GP in some cases. The fact that BALLET-RTS uniformly outperforms LA-MCTS and TuRBO-DK on all the experiments requires further study on integrating Thompson sampling into a BALLET-style framework with a similar theoretical guarantee.

5. Conclusion

We propose a novel framework for adaptively learning regions of interest for Bayesian optimization. Our model maintains two Gaussian processes: One global model for identifying the ROIs as (adaptive) superlevel-sets; the other surrogate model for acquiring data in these high-confidence ROIs. We proposed to take the width of the intersection of the point-wise confidence intervals of both GPs as the acquisition function to achieve a theoretical guarantee on both the convergence rate of the filtering and optimization process. We demonstrate our algorithm in promising real-world experiment design scenarios, including protein engineering and material science. Our results show that BALLET compares favorably against state-of-the-art BO approaches under similar settings—especially in high-dimensional and structured tasks with non-stationary dynamics—while having fewer hyperparameters to fine-tune.

We show the potential of integrating Thompson sampling into the framework, and the extensions to other acquisition functions that are not based on confidence intervals are also of interest for future work. We demonstrate the practical issues of taking the intersection of all historical CIs and discuss the cost of only taking the intersection of CIs at each time step. This raises the demand for future studies on addressing the dynamics of (deep) kernel learning.

References

Barlow, K. A., O Conchuir, S., Thompson, S., Suresh, P., Lucas, J. E., Heinonen, M., and Kortemme, T. Flex ddg: Rosetta ensemble-based estimation of changes in protein–protein binding affinity upon mutation. *The Journal of Physical Chemistry B*, 122(21):5389–5399, 2018. 6

Bengio, Y., Delalleau, O., and Le Roux, N. The curse of dimensionality for local kernel machines. *Techn. Rep*, 1258:12, 2005. 1

Berkenkamp, F., Schoellig, A. P., and Krause, A. Safe controller optimization for quadrotors with Gaussian processes. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 493–496, 2016. 1

Bogunovic, I., Scarlett, J., Krause, A., and Cevher, V. Truncated variance reduction: A unified approach to bayesian optimization and level-set estimation. *Advances in neural information processing systems*, 29, 2016. 3

Chapelle, O. and Li, L. An empirical evaluation of thompson sampling. *Advances in neural information processing systems*, 24, 2011. 6

Das, R. and Baker, D. Macromolecular modeling with rosetta. *Annu. Rev. Biochem.*, 77:363–382, 2008. 6

Desautels, T., Zemla, A., Lau, E., Franco, M., and Faissol, D. Rapid in silico design of antibodies targeting sars-cov-2 using machine learning and supercomputing. *BioRxiv*, 2020. 7

Djolonga, J., Krause, A., and Cevher, V. High-dimensional gaussian process bandits. *Advances in neural information processing systems*, 26, 2013. 2

Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., and Poloczek, M. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems*, pp. 5496–5507, 2019. 1, 2, 3, 6, 12

Ferreira, M. F., Camacho, R., and Teixeira, L. F. Using autoencoders as a weight initialization method on deep neural networks for disease detection. *BMC Medical Informatics and Decision Making*, 20(5):1–18, 2020. 3

Gotovos, A., Casati, N., Hitz, G., and Krause, A. Active learning for level set estimation. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pp. 1344–1350. AAAI Press, 2013. ISBN 9781577356332. 1, 3

Kawaguchi, K., Maruyama, Y., and Zheng, X. Global continuous optimization with error bound and fast convergence. *Journal of Artificial Intelligence Research*, 56:153–195, 2016. 3

Kirschner, J., Mutny, M., Hiller, N., Ischebeck, R., and Krause, A. Adaptive and safe bayesian optimization in high dimensions via one-dimensional subspaces. In *International Conference on Machine Learning*, pp. 3429–3438. PMLR, 2019. 2

Makarova, A., Usmanova, I., Bogunovic, I., and Krause, A. Risk-averse heteroscedastic bayesian optimization. *Advances in Neural Information Processing Systems*, 34, 2021. 3

Merrill, E., Fern, A., Fern, X., and Dolatnia, N. An empirical study of bayesian optimization: Acquisition versus partition. *Journal of Machine Learning Research*, 22(4): 1–25, 2021. 3

Munos, R. Optimistic optimization of a deterministic function without the knowledge of its smoothness. *Advances in neural information processing systems*, 24, 2011. 3

Munos, R. *From Bandits to Monte-Carlo Tree Search: The Optimistic Principle Applied to Optimization and Planning*. now, 2014. 1, 3

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2006. 1, 3

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021. 6

Salgia, S., Vakili, S., and Zhao, Q. A domain-shrinking based bayesian optimization algorithm with order-optimal regret performance. *Advances in Neural Information Processing Systems*, 34:28836–28847, 2021. 2

Sazanovich, M., Nikolskaya, A., Belousov, Y., and Shpilman, A. Solving black-box optimization challenge via learning search space partition for local bayesian optimization. In Escalante, H. J. and Hofmann, K. (eds.), *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*, volume 133 of *Proceedings of Machine Learning Research*, pp. 77–85. PMLR, 06–12 Dec 2021. 2, 3

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104 (1):148–175, 2016. doi: 10.1109/JPROC.2015.2494218. 3

Snoek, J., Larochelle, H., and Adams, R. P. Practical bayesian optimization of machine learning algorithms. In *26th Annual Conference on Neural Information Processing Systems 2012*, pp. 2951–2959, 2012. 1

Song, J., Tokpanov, Y., Chen, Y., Fleischman, D., Fountaine, K., Atwater, H., and Yue, Y. Optimizing photonic nanostructures via multi-fidelity gaussian processes. *NeurIPS Workshop on Machine Learning for Molecules and Materials*, 2018. 6

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009. 1, 5, 11, 12

Sui, Y., Zhuang, V., Burdick, J., and Yue, Y. Stagewise safe bayesian optimization with gaussian processes. In *International conference on machine learning*, pp. 4781–4789. PMLR, 2018. 1, 3

Wabersich, K. P. and Toussaint, M. Advancing bayesian optimization: The mixed-global-local (mgl) kernel and length-scale cool down. *arXiv preprint arXiv:1612.03117*, 2016. 2

Wang, L., Fonseca, R., and Tian, Y. Learning search space partition for black-box optimization using monte carlo tree search. *Advances in Neural Information Processing Systems*, 33:19511–19522, 2020. 2, 3, 6, 12

Wang, Z. and Jegelka, S. Max-value entropy search for efficient bayesian optimization. In *International Conference on Machine Learning*, pp. 3627–3635. PMLR, 2017. 1

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and de Feitas, N. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016a. 2

Wang, Z., Zhou, B., and Jegelka, S. Optimization as estimation with gaussian processes in bandit settings. In *Artificial Intelligence and Statistics*, pp. 1022–1031. PMLR, 2016b. 1

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. Deep kernel learning. volume 51 of *Proceedings of Machine Learning Research*, pp. 370–378, Cadiz, Spain, 09–11 May 2016. PMLR. 3, 14

Wistuba, M. and Grabocka, J. Few-shot bayesian optimization with deep kernel surrogates. *arXiv preprint arXiv:2101.07667*, 2021. 3, 14

Wu, Z., Kan, S. J., Lewis, R. D., Wittmann, B. J., and Arnold, F. H. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proceedings of the National Academy of Sciences*, 116(18):8852–8858, 2019. 6

Yang, K. K., Wu, Z., and Arnold, F. H. Machine-learning-guided directed evolution for protein engineering. *Nature methods*, 16(8):687–694, 2019. 1

Zhang, F., Nord, B., and Chen, Y. Learning representation for bayesian optimization with collision-free regularization. *arXiv preprint arXiv:2203.08656*, 2022. 1, 6, 12

A. Proofs

A.1. Proof of Lemma 1 and Corollary 1

Proof: Similar to lemma 5.1 of (Srinivas et al., 2009), with probability at least $1-\delta$, $\forall \mathbf{x} \in \tilde{D}, \forall t \geq 1, \forall f \in \{f_g, \hat{f}\}$,

$$|f(\mathbf{x}) - \mu_{f,t-1}(\mathbf{x})| \leq \beta_t^{1/2}\sigma_{f,t-1}(\mathbf{x})$$

Note that we also take the union bound on $f \in \{f_g, \hat{f}\}$.

Then $\forall t \geq 1, \forall f \in \{f_g, \hat{f}\}$,

$$P\left(f^* \leq \text{UCB}_{f,t}(\mathbf{x}^*) \leq \text{UCB}_{f,t,\max}\right) \geq 1-\delta$$

According to equation 5, $\forall t \geq 1$

$$P\left(f^* \leq \max_{\mathbf{x}\in\mathbf{X}} \widehat{\text{UCB}}_t(x)\right) \geq 1-\delta$$

Symmetrically, $\forall \mathbf{x} \in \tilde{D}, \forall t \geq 1, \forall f \in \{f_g, \hat{f}\}$,

$$P\left(f^* \geq f(\mathbf{x}) \geq \text{LCB}_{f,t}(\mathbf{x})\right) \geq 1-\delta$$

Then $\forall t \geq 1$,

$$P\left(\text{UCB}_{f_g,t}(\mathbf{x}) \geq f^* \geq \text{LCB}_{f_g,t,\max}\right) \geq 1-\delta$$

according to the definition of $\hat{\mathbf{X}}$, $P\left(\mathbf{x}^* \in \hat{\mathbf{X}}_t\right) \geq 1-\delta$.

Also, according to equation 5, $\forall t \geq 1$

$$P\left(f^* \geq \max_{\mathbf{x}\in\mathbf{X}} \widehat{\text{LCB}}_t(x)\right) \geq 1-\delta$$

□

A.2. Proof of Proposition 1

The following two lemmas shows that the width of the interval is bounded by the maximum of $\alpha_{\hat{f}}$.

Lemma 2 *Under the same conditions assumed in Lemma 1 except for* $\beta_t = 2\log(2|\tilde{D}\cap\hat{\mathbf{X}}|\pi_t/\delta)$, *with acquisition function* $\alpha_{\hat{f}}(\mathbf{x}) = |\widehat{CI}_t(\mathbf{x})|$, $\forall t \geq 1, \forall f \in \{f_g, \hat{f}\}$, *let* $\mathbf{x}'' = \arg\max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} |\widehat{CI}_t(\mathbf{x})|$ *we have* $\max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{UCB}_t(\mathbf{x}) - \max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{LCB}_t(\mathbf{x}) \leq \rho_{CI}\beta_t^{1/2}\sigma_{f,t-1}(\mathbf{x})$. *Here* $\rho_{CI} \leq \rho_{UCB} \leq 2$.

Proof: $\forall t \geq 1, \forall f \in \{f_g, \hat{f}\}$

$$\begin{aligned}
\max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{\text{UCB}}_t(\mathbf{x}) - \max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{\text{LCB}}_t(\mathbf{x}) &\leq \widehat{\text{UCB}}_t(\mathbf{x}') - \widehat{\text{LCB}}_t(\mathbf{x}') \\
&\leq 2\beta_t^{1/2}\sigma_{f,t-1}(\mathbf{x}') \\
&\leq \arg\max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} |\widehat{\text{CI}}_t(\mathbf{x})| \\
&= 2\beta_t^{1/2}\sigma_{f,t-1}(\mathbf{x}'')
\end{aligned}$$

□

The followings finish the proof of Proposition 1.

Proof: By lemma 5.4 of Srinivas et al. (2009), with $\beta_t = 2\log(2|\tilde{D}\cap\hat{\mathbf{X}}|\pi_t/\delta)$, $\forall f \in \{f_g, \hat{f}\}$, $\sum_{t=1}^{T}(2\beta_t^{1/2}\sigma_{f,t-1}(\mathbf{x}_t))^2 \leq C_1\beta_T\gamma_{f,T}$. Taking the union bound of Lemma 1 and Corollary 2, with probability at least $1-2\delta$, $\forall f \in \{f_g, \hat{f}\}$,

$$\begin{aligned}
\sum_{t=1}^{T}\left(\max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{\text{UCB}}_t(\mathbf{x}) - \max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{\text{LCB}}_t(\mathbf{x})\right)^2 &\leq \sum_{t=1}^{T}(\rho_{\alpha_{\hat{f}}}\beta_t^{1/2}\sigma_{f,t-1}(\mathbf{x}_t))^2 \\
&\leq \rho_{\alpha_{\hat{f}}}^2 C_1\beta_T\gamma_{f,T}/4
\end{aligned}$$

According to equation 5, $\max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{\mathrm{UCB}}_t(\mathbf{x}) - \max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{\mathrm{LCB}}_t(\mathbf{x})$ is monotonically decreasing. By Cauchy-Schwaz, with probability at least $1-2\delta, \forall f \in \{f_g, \hat{f}\}$,

$$\begin{aligned}
\rho^2_{\alpha_{\hat{f}}} C_1 \beta_T \gamma_{f,T}/4 &\geq \sum_{t=1}^{T} \left(\max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{\mathrm{UCB}}_t(\mathbf{x}) - \max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{\mathrm{LCB}}_t(\mathbf{x}) \right)^2 \\
&\geq \frac{1}{T} (\sum_{t=1}^{T} \max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{\mathrm{UCB}}_t(\mathbf{x}) - \max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{\mathrm{LCB}}_t(\mathbf{x}))^2 \\
&\geq T \left(\max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{\mathrm{UCB}}_T(\mathbf{x}) - \max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{\mathrm{LCB}}_T(\mathbf{x}) \right)^2
\end{aligned}$$

Assume with probability at least $1-2\delta$,

$$\left(\max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{\mathrm{UCB}}_T(\mathbf{x}) - \max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{\mathrm{LCB}}_T(\mathbf{x}) \right)^2 \leq \rho^2_{\alpha_{\hat{f}}} C_1 \beta_T \widehat{\gamma}_T / 4T \leq \epsilon^2$$

Hence, with the smallest T satisfying $T \geq \frac{\rho^2_{\alpha_{\hat{f}}} \beta_T \widehat{\gamma}_T C_1}{4\epsilon^2}$, $\mathbb{P}\left[\max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{\mathrm{UCB}}_T(\mathbf{x}) - \max_{\mathbf{x}\in\tilde{D}\cap\hat{\mathbf{X}}} \widehat{\mathrm{LCB}}_T(\mathbf{x}) \leq \epsilon\right] \geq 1-2\delta$.
□

B. Discussions

Smoothness improvement on ROI In near-optimal areas, the smoothness of the objective should be no worse than the smoothness in the larger (global) area. (Srinivas et al., 2009) discussed the role of smoothness in reducing γ. As indicated by Proposition 1, the benefits to optimization of a smoother kernel learned on ROI instead of the kernel learned on the globe could be reflected in the reduced $\widehat{\gamma}$ in the regret bound compared to γ_{f_g} without the filtering of BALLET.

C. Supplemental Experimental Results

In this section, we include an extended empirical study of BALLET, compared against a broader collection of baseline algorithms with varying hyperparameters. Specifically, we show the results for the following algorithms:

- *BALLET-ICI-RBF*: BALLET-ICI with RBF (squared-exponential) base kernel $k(\mathbf{x},\mathbf{x}') = \exp\left(-\frac{d(\mathbf{x},\mathbf{x}')^2}{2l^2}\right)$.
- *BALLET-ICI-Lin*: BALLET-ICI with linear base kernel $k(\mathbf{x},\mathbf{x}') = \sigma_0^2 + \mathbf{x}\cdot\mathbf{x}'$ (with prior $N(0,\sigma_0^2)$ on the bias).
- *DKBO-AE-RBF.* DKBO-AE with RBF base kernel (Zhang et al., 2022).
- *DKBO-AE-Lin.* DKBO-AE with linear base kernel.
- LA-MCTS. The Latent Action Monte Carlo Tree Search algorithm (LA-MCTS) of Wang et al. (2020).
- *TuRBO-m.* The Trust region Bayesian optimization (TuRBO) algorithm of Eriksson et al. (2019), where m specifies the variant of TuRBO that maintains m local models in parallel.

As shown in table 1, BALLET-ICI (with different choices of base kernels) consistently outperforms other baselines on all datasets but Nanophotonics. On Nanophotonics, while there is a small gap between BALLET-ICI and LA-MCTS, LA-MCTS is relatively unstable with a larger variance (SE). This is consistent with the results reported in figure 3.

Hyperpameter choice We further provide results on BALLET's performance with varying β when filtering. Figure 4 shows the simple regret of BALLET-ICI on the *Nanophotonics* dataset. We observe that—although our regret bounds in section 3.2 rely on specific choices of $\beta_t^{1/2}$ for filtering – the empirical results are robust within a range of small values. Also, using the $\beta_T^{1/2} = 6.2$ as the analytic results in Proposition 1 failed to match the performance of the fixed $\beta_t^{1/2} \leq 1$, showing its over-conservative problem.

	1-D toy	**HDBO**	**Nanophotonics**	**WaterConverter**	**GB1**	**Rosetta**
	$T = 40$	$T = 40$	$T = 90$	$T = 90$	$T = 90$	$T = 90$
BALLET-ICI-RBF	$\mathbf{0.03 \pm 0.01}$	$\mathbf{85.90 \pm 7.29}$	76.65 ± 9.55	33664.62 ± 0.00	$\mathbf{4.81 \pm 0.15}$	$\mathbf{3.86 \pm 0.49}$
BALLET-ICI-Lin	0.10 ± 0.03	110.81 ± 7.93	34.49 ± 4.39	$\mathbf{20084.66 \pm 2928.84}$	6.33 ± 0.28	5.11 ± 0.18
DKBO-AE-RBF	0.05 ± 0.02	90.75 ± 16.01	89.49 ± 3.44	28591.49 ± 2560.23	5.02 ± 0.41	4.89 ± 0.16
DKBO-AE-Lin	0.07 ± 0.01	92.84 ± 6.22	82.94 ± 4.50	33664.63 ± 0	6.44 ± 0.19	4.12 ± 0.46
LA-MCTS	0.10 ± 0.04	95.47 ± 4.84	$\mathbf{30.79 \pm 10.28}$	26814.43 ± 1593.76	5.59 ± 0.40	5.09 ± 0.32
TuRBO-1	0.31 ± 0.00	136.80 ± 0.00	96.58 ± 0.00	33664.69 ± 0.00	5.34 ± 0.52	6.67 ± 0.00
TuRBO-2	0.07 ± 0.06	105.51 ± 5.47	50.60 ± 15.46	28450.65 ± 1691.96	4.95 ± 0.45	5.75 ± 0.32
TuRBO-4	0.04 ± 0.03	93.74 ± 10.25	63.60 ± 3.48	32800.93 ± 3148.98	5.72 ± 0.72	5.46 ± 0.22

Table 1: Simple regret (Mean ± SE) at the T^{th} iteration on the 6 datasets described in section 4. Here, T aligns with the optimization horizon reported in figure 3 for each dataset. The top results are highlighted in bold.

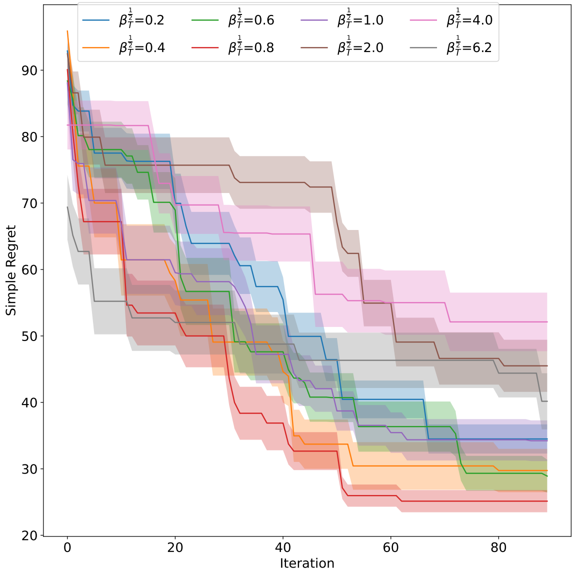


Figure 4: Effect of scaling parameter β on the *Nanophotonics* dataset. For $\beta_T^{1/2} \leq 4$, the values are fixed through the optimization, while $\beta_T^{1/2} = 6.2$ corresponds to the results of varying $\beta_t^{1/2}$ as in Proposition 1.

D. Additional Results

D.1. TRUVAR Results

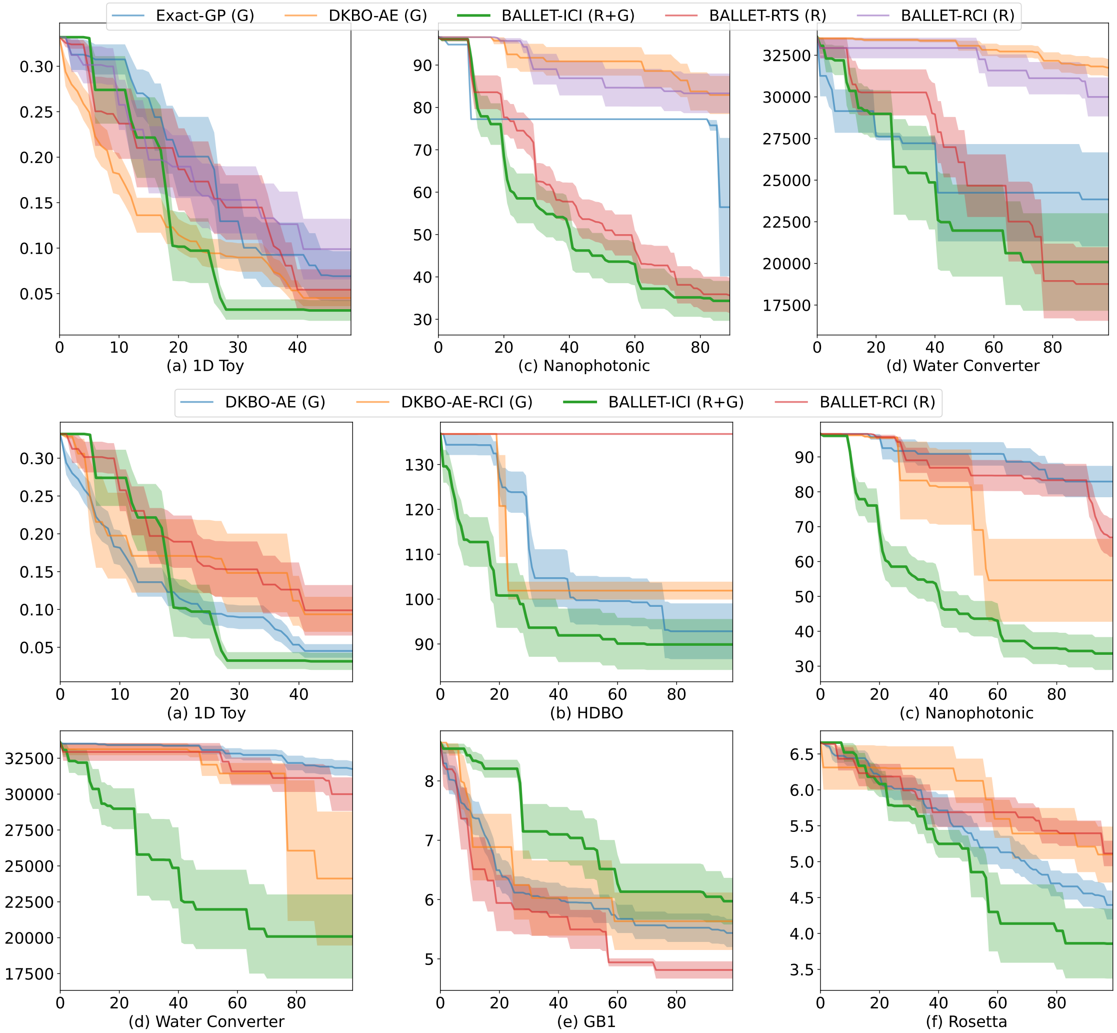
We do not include TRUVAR in the main paper for the following reasons. (1) TRUVAR is **not a partition-based BO method** that aims at resolving the heteroscedasticity in BO by learning local models; (2) It is prohibitive to run for large candidate sets as TRUVAR's acquisition function requires estimating the posterior variance reduction for all the remaining candidates. We observe on the 1D-toy dataset the simple regret is 0.121 ± 0.033 by TRUVAR v.s. 0.0031 ± 0.011 by BALLET-ICI.

D.2. Exact-GP results

We Compare Exact-GP results on 1-D Toy, Nanophotonics, and Water converter configuration datasets with DKBO-AE as an ablation study of deep kernel learning. The choice of kernels and hypereparameters are identical to the deep kernel discussed in section 4 except for removing the latent space mapping and kernel interpolation. As is shown in figure 5, Exact-GP is consistently outperformed by DKBO-AE and BALLET-ICI.

D.3. RCI results

We compare DKBO-AE-RCI directly with DKBO-AE as the direct ablation study of the proposed acquisition function. The choice of kernels and hypereparameters are identical to the deep kernel discussed in section 4. The acquisition function $\text{UCB}_{f_g,t}(\mathbf{x}) - \text{LCB}_{f_g,t}$, which is similar to equation 9, is maximized over $\mathbf{X}$ instead of $\hat{\mathbf{X}}$.



(a) 1D Toy

(c) Nanophotonic

(d) Water Converter

(a) 1D Toy

(b) HDBO

(c) Nanophotonic

(d) Water Converter

(e) GB1

(f) Rosetta

Figure 5: Simulation results on each task are shown here. The error bar demonstrates the standard error. The x-axis denotes the number of iterations, and the y-axis denotes the simple regret. The simple regrets for the 10 initial randomly picked warm-up datasets are clipped. (G), (R), and (R+G) means the global model only, the ROI model only, and the ROI model combined with the global model correspondingly.

As is shown figure 5, BALLET-ICI outperforms the baselines except on GB1, indicating the advantage of leveraging both global and local information together. BALLET-RCI performs the best and DKBO-AE-RCI outperforms BALLET-ICI on GB1. This shows the benefits of identifying the ROI and optimizes on it, and the harm a potential discrepancy between the global model and the ROI model could be to the optimization.

E. Discussions

E.1. Computational Cost

In deep kernel learning, which is shown to bear strong empirical performance in regression and optimization (e.g.(Wilson et al., 2016; Wistuba & Grabocka, 2021)), the learning cost is $\mathcal{O}(n)$ for n training points, and the prediction cost is $\mathcal{O}(1)$ per test point and is more efficient than the exact GP in terms of computational cost. Compared to the **significant experiment cost** in the real-world application BALLET is proposed for (e.g., cosmological design, protein study), the computational cost is negligible. Meanwhile, the runtime of other partition-based algorithms depends on the **hyperparameters** of the partitioning heuristics, e.g., K-means iterations in LA-MCTS, the number and size of trust regions in TuRBO.

E.2. Limitation and Future Work

We summarize the following limitations throughout the paper.

- The analysis only applies to given discretization, while sampling and related work focus on the issue;
- It Doesn't help to learn an ROI GP when the objective has global uniformity. The global kernel itself forms a good surrogate GP.
- The analysis should be able to extend to more acquisition functions.
- Lack of analysis on top of the (deep) kernel learning. Though different from applying an exact GP through the optimization process, deep kernel learning has shown strong performance in regression and optimization tasks. The gap between DK-based BO and exact GP-based BO remains to be filled.