Designing Search Space for Unbounded Bayesian Optimization via Transfer Learning

Quoc-Anh Hoang Nguyen[*1], Hung The Tran[*3], Sunil Gupta[2], and Dung D. Le[3] (✉)

[1] FPT Software AI Center, Vietnam, nhquocanh@gmail.com
[2] Deakin University, Australia, sunil.gupta@deakin.edu.au
[3] College of Engineering and Computer Science, Center of Environmental Intelligence, VinUniversity, Vietnam,
tran.thehung1705@gmail.com,Dung.ld@vinuni.edu.vn

Abstract. Bayesian optimization (BO) is a powerful method for optimizing expensive black-box functions and has been successfully applied across various scenarios. While traditional BO algorithms optimize each task in isolation, there has been recent interest in speeding up BO by transferring knowledge across similar previous tasks. However, most recent studies on this problem are based on two implicit assumptions that (1) the search space of the test task (the ultimate task the model aims to solve) needs to be defined suitably a priori and (2) the optimum of the test task is very close to the evaluations of the previous tasks. These restrictive assumptions limit BO's applicability in real-world scenarios. In this paper, we propose an approach that leverages transfer learning to design promising search spaces for BO, thereby overcoming these limitations. Our approach eliminates the need for prior knowledge of the search spaces of both the test and previous tasks while also relaxing the assumption that the test task's optimum is close to evaluations of previous tasks. We propose a novel BO algorithm to automatically design promising search spaces for BO, not only exploiting regions near good evaluations of previous tasks but also exploring other promising regions using strategy shifting and expanding the search space. Our algorithm leverages both task similarity measurements and the best evaluation achieved so far for the test task. Further, theoretically, we prove that our proposed algorithm is guaranteed to find a global optimum in the worst-case scenario although the search spaces are unknown. Finally, we empirically demonstrate that our algorithms considerably boost BO and outperform the state-of-the-art on a wide range of benchmarks.

Keywords: Bayesian optimization · Transfer Learning · Gaussian Process · Designing search space.

1 Introduction

Bayesian optimization (BO) is a powerful method for optimizing expensive black-box functions. It works by iteratively fitting a surrogate model, usually a Gaus-

*These authors contributed equally to this work.

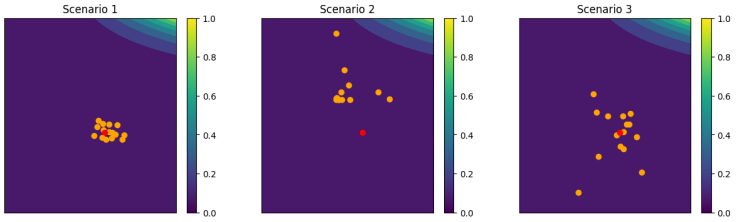

Fig. 1. True contour plot of the Beale 2D-function and data distribution on 3 scenarios. The red dots: the global optimum of the target task; the yellow dots: the optimum point of training tasks. (**Left**) Target task optimum is close to most training tasks' optima; (**Middle**) Target task optimum is distant from most training tasks' optima; (**Right**) Target task optimum is distant from some training tasks' optima.

sian process (GP), and maximizing an acquisition function to determine the next evaluation point. Bayesian optimization algorithms have proven particularly successful in a wide variety of domains including hyperparameter tuning [2], reinforcement learning [14], neural architecture search [11], and Pareto front learning [29, 30].

However, two issues hamper the efficiency of BO in real-world applications are (1) **slow convergence:** given a very limited budget, BO methods often fail to converge to a good solution quickly [1] and **search space definition:** Traditional BO requires the user to define a suitable search space a priori. However, defining a default search space for a particular data mining problem is difficult and left to human experts [17]. For example, in many machine learning algorithms, the hyperparameters or parameters can take values in an unbounded space e.g. L_1/L_2 penalty hyperparameters in elastic-net can take any non-negative value; To address the first issue, transfer learning is an efficient solution to speed up BO by leveraging the information obtained from similar previous tasks into optimization. In many practical applications, optimizations are repeated in similar settings. Examples include hyperparameter optimization, which is repeatedly done for the same machine learning model on varying datasets, or the optimization of control parameters for a given system with varying physical configurations.

Most of the recent works for this transfer learning-based search space design (e.g., [17, 12]) are based on an implicit hypothesis that the optimums of an objective function (test task) are close to the best evaluations of the previous tasks (or training tasks). Figure 1 (Left) shows an example of this situation. As a result, they search only regions close to those evaluations. [17] uses a quite simple strategy. Instead of searching on the whole search space, they learn a region in the form of a box or an ellipsoid bounding the best evaluations of training tasks. However, if the distribution of these best evaluations is large then the learned search space is not improved compared to the original search space. This is illustrated in Figure 1 (Right). Another recent work [12] proposes to use a similarity measurement to choose the good training tasks that are similar

to the test task. Then, they use a Gaussian process classifier (GPC) to design promising regions, where the classifier predicts whether a point from the search space belongs to the promising region or not. However, this approach has two limitations. First, it is designed for the search space which is discrete and known. If the search space is continuous, classifying each point from the search space into a class is expensive or even intractable in the unknown search space setting. Second, the similarity measurement used in [12] is efficient only when the test task has enough evaluations and the distribution of evaluations in the test task is uniform. We will discuss this in detail in Section 5.1.1.

In practice, the optimums of the test task may not be close to the best evaluations of previous tasks as illustrated in Figure 1 (Middle). Consequently, both works [17, 12] may encounter limitations in such scenarios. Furthermore, all these approaches rely on prior knowledge of the search space, and none address the above second issue, which is challenging.

In this paper, we address designing promising search spaces with the merit of transfer learning to speed up BO (reducing the number of evaluations) without requiring prior knowledge of the search space. We propose a novel algorithm to learn promising search spaces for BO, which not only exploits regions close to good evaluation points of previous tasks, but also explores promising regions surrounding the best evaluations of the test task. We propose using a strategy of shifting and expanding the search space and a novel similarity measurement. Furthermore, theoretically, we prove that our proposed algorithm is guaranteed to find a global optimum in the worst-case scenario, even when the search spaces are unknown. Finally, we demonstrate that our proposed algorithm considerably boosts BO, and outperforms the state-of-the-art on a wide range of benchmarks.

2 Related works

Previous work has implemented transfer learning for BO in different ways to leverage the auxiliary information from similar training tasks to achieve faster optimization on the target task. One way is to learn surrogate models from source tasks. For example, [27] and [7] train individual surrogate models on each dataset and then combine them using a weighted sum-based approach. Another approach, proposed by [13], uses a two-phase framework to extract and aggregate knowledge from both source and target tasks. [20] employ neural networks to learn basis functions for Bayesian linear regression. Furthermore, several works consider the difference between training tasks and target tasks to compute the kernel of the proposed surrogate [22, 28]. The other line of transfer learning works focuses on designing acquisition functions. [27] introduce a TAF that utilizes a variant of the EI acquisition function to leverage the improvement of new points. [4] propose a method called RM-GP-UCB, where the acquisition function is a weighted combination of individual GP-UCB acquisition functions for both the target task and the training tasks. [26] employ reinforcement learning to meta-train an acquisition function on a set of related tasks, allowing the incorporation of implicit structural knowledge. From a few-shot learning perspective, [10] pro-

pose FSFA, which effectively adapts to a wide range of black-box functions using a small amount of meta-data. In contrast, our work focuses on the design of the search spaces for transfer learning in BO, taking an orthogonal direction to the aforementioned methods. It is worth noting that our method can be combined with these transfer learning-based approaches to enhance the efficiency of BO. [6] addressed BO using transfer learning where the search spaces of the task and previous tasks are heterogeneous.

BO with unknown search spaces has been considered in previous works. [16] consider the weakly specified search space for BO and propose the filtering expansion strategy. This approach is reasonable when a training dataset is available, as it allows for the initial region to be located near the best evaluation of the training tasks. Nevertheless, maximizing the acquisition function within their expanded search space is challenging since the invasion set needs to be specified. [9] provide the search space expansion strategy to achieve the $\epsilon-$ accuracy after a finite number of iterations. [3] proposes an adaptive expansion strategy based on the uncertainty of GP model. [23, 19] increase the volume of the search space to guarantee to contain the global optimum of an objective function. However, none of these works consider transfer learning to leverage data from the previous training tasks. To our knowledge, we are the first to consider transfer learning for BO with unknown search spaces by novel expansion strategies.

[5] propose a trust-region method called TuRBO, which is an effective BO method for high-dimensional problems. Their method is based on adjusting the size of the trust region and moving towards the best solution so far. Our proposed methods also use the adjustment and the movement of boxes but with novel strategies by integrating transfer learning. In addition, their method is nearly a local strategy without providing a convergence analysis. In contrast, we demonstrate theoretically that our global method converges sub-linearly.

3 Preliminaries

Bayesian optimization (BO) finds the global optimum of an unknown, expensive, possibly non-convex function $f(x)$. It is assumed that we can interact with f only by querying at some $x \in \mathbb{R}^d$ and obtain a noisy observation $y = f(x) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. The search space is required to be specified a priori and is assumed to include the true global optimum. BO proceeds sequentially in an iterative fashion. At each iteration, a surrogate model is used to probabilistically model $f(x)$. Gaussian process (GP) [18] is a popular choice for the surrogate model as it offers a prior over a large class of functions and its posterior and predictive distributions are tractable. Formally, we have $f(x) \sim \mathcal{GP}(m(x), k(x, x'))$ where $m(x)$ and $k(x, x')$ are the mean and the covariance (or kernel) functions. Popular covariance functions include Squared Exponential (SE) kernels, Matérn kernels, etc. Given a set of observations $\mathcal{D}_{1:t} = \{x_i, y_i\}_{i=1}^t$, the predictive distribution can be derived as $P(f_{t+1}|\mathcal{D}_{1:t}, x) = \mathcal{N}(\mu_{t+1}(x), \sigma_{t+1}^2(x))$, where $\mu_{t+1}(x) = \mathbf{k}^T[\mathbf{K} + \sigma^2\mathbf{I}]^{-1}\mathbf{y} + m(x)$ and $\sigma_{t+1}^2(x) = k(x, x) - \mathbf{k}^T[\mathbf{K} + \sigma^2\mathbf{I}]^{-1}\mathbf{k}$. In the above expression we define $\mathbf{k} = [k(x, x_1), ..., k(x, x_t)]$, $\mathbf{K} = [k(x_i, x_j)]_{1 \le i,j \le t}$ and $\mathbf{y} = [y_1, \ldots, y_t]$.

After the modeling step, an acquisition function is used to suggest the next x_{t+1} where the function should be evaluated. The acquisition step uses the predictive mean and the predictive variance from the surrogate model to balance the exploration of the search space and exploitation of current promising regions. Some examples of acquisition functions include Expected Improvement (EI) [15, 24], GP-UCB [21, 25]. In this paper, we use the UCB acquisition function which is defined as follows:

$$u_t(x) = \mu_{t-1}(x) + \beta_t^{1/2}\sigma_{t-1}(x), \tag{1}$$

where β_t is the trade-off coefficient that balances exploration and exploitation.

To measure the performance of a BO algorithm, we use the regret, which is the loss incurred by evaluating the function at x_t, instead of at the unknown optimal input, formally $r_t = f(x^*) - f(x_t)$. The cumulative regret is defined as $R_T = \sum_{1 \leq t \leq T} r_t$, the sum of regrets incurred over given a horizon of T iterations. If we can show that $lim_{T\to\infty}\frac{R_T}{T} = 0$, the cumulative regret is **sub-linear**, and so the algorithm efficiently converges to the optimum.

4 Problem Setting

The goal of Bayesian optimization is to find a maximum of the objective function $f(x)$: $\text{argmax}_{x\in\mathbb{R}^d} f(x)$. We consider the function f that is black-box and expensive to evaluate, possibly non-convex. Further, we only get access to noisy evaluations of f without gradient information in the form $y = f(x) + \epsilon$, where the noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is i.i.d. Gaussian distribution. Unlike traditional BO, we assume that the search space $\mathcal{X} \subset \mathbb{R}^d$ of f is unknown a priori. As in [9], we assume that x^* is not at infinity to make the BO tractable.

Without the knowledge of the search space, BO is challenging. However, we assume that we have knowledge of K previous related BO tasks $\{f^{(k)}(x)\}_{k=1}^K$, where $f^{(k)} : \mathbb{R}^d \to \mathbb{R}$ has the same input dimension of f. More precisely, we have access to noisy observations from these tasks, which are denoted by $\mathcal{D}^{(k)} = \{x_i^{(k)}, y_i^{(k)} = f^{(k)}\left(x_i^{(k)}\right) + \epsilon\}_{i=1}^{n_k}$, where n_k is the number of observations of the task k. We create a compact search space $\hat{\mathcal{X}}$, given noisy observations from previous BO tasks for the following problem:

$$\text{argmax}_{x\in\hat{\mathcal{X}}} f(x) \tag{2}$$

The smaller the search space $\hat{\mathcal{X}}$ is, the faster the optimization methods may find the optimum of that space. Therefore, we aim to design a small $\hat{\mathcal{X}}$ so that it contains an optimum of the original space $\mathcal{X}$.

5 Designing search spaces for BO

In this section, we propose a safety transfer learning search space strategy that guarantees the containment of the global optimum of the target task after finite steps without needing to know the original search space. Previous transfer

Algorithm 1 Designing search spaces for Bayesian optimization

1: Initial search space $\mathcal{X}_0$; Set of initial points in $\mathcal{X}_0$, denoted by $\mathcal{D}_0$; $\epsilon > 0$; $m > 0$.
2: **for** $t = 1, 2, ...$ **do**
3: Fit the Gaussian process using $\mathcal{D}_{t-1}$.
4: Define $\hat{\mathcal{X}}_t$ using (3).
5: Find $x_t = \text{argmax}_{x \in \hat{\mathcal{X}}_t} u_t(x)$, where $u_t(x)$ defined as in Eq (1) to find x_t.
6: Sample $y_t = f(x_t) + \epsilon_t$.
7: Augment the data $\mathcal{D}_t = \{\mathcal{D}_{t-1}, (x_t, y_t)\}$.
8: **end for**

learning space works [12, 17] have not addressed this safety concern (containment of global optimum), even within bounded settings. For our approach, the intuition is that starting with the small good initial region, we will simultaneously move and expand the search space so that it can reach the more promising region. The moving strategy will take advantage of transfer knowledge so that it can exploit efficiency in the initial stages while the expanding strategy will ensure the search space contains the global optimum, solving the above safety issue. Specifically, starting from a good initial user-defined region, denoted by $\hat{\mathcal{X}}_0 = [a_0^{(1)}, b_0^{(1)}] \times ... \times [a_0^{(d)}, b_0^{(d)}]$, the search space at iteration t, denoted by $\hat{\mathcal{X}}_t = [a_t^{(1)}, b_t^{(1)}] \times ... \times [a_t^{(d)}, b_t^{(d)}]$ will be built from $\hat{\mathcal{X}}_{t-1}$ by a sequence of transformations as follows:

$$\hat{\mathcal{X}}_{t-1} \xrightarrow{\text{move}} \hat{\mathcal{X}}'_t \xrightarrow{\text{expand}} \hat{\mathcal{X}}_t \tag{3}$$

Our algorithm is described in Algorithm 1. We will detail the expanding and moving strategy below.

5.1 Moving Strategy

A promising region of a task is where we have a belief that it contains an optimum of the task with a high probability. Our intuition is that when the target task is similar to a training task by some measurement, promising regions are similar in both tasks, so regions surrounding the best evaluations of the training task are potentially promising regions of the test task to be exploited. Therefore, we can effectively exploit these regions in the early stages.

Similarity Measurement To measure the task similarity between source tasks and the target task, [8, 12] use the Kendall tau rank correlation coefficient:

$$L(f^{(k)}, f|\mathcal{D}_t) = \frac{\Sigma_{x_i, x_j \in \mathcal{D}_t} \mathbb{I}\left[\left(M^{(k)}(x_i) < M^{(k)}(x_j)\right) \otimes (y_i < y_j)\right]}{t(t-1)}, \tag{4}$$

where $M^{(k)}(.)$ denotes the mean of GP trained on training dataset $\{x_i^{(k)}, y_i^{(k)}\}_{i=1}^{n_k}$, $\mathcal{D}_t = \{x_i, y_i\}_{i=1}^t$ is the observations of target task at iteration t, and $\otimes$ denotes the exclusive NOR operator, e.g value is true if the two sub-statements return

the same value. Intuitively, the numerator of Equation 4 counts the number of true ranked pairs between GP trained on training dataset $\mathcal{D}^{(k)}$ and the observations of the target task. The ranking is more reasonable than other choices such that squared error as the focus lies solely on identifying promising regions. However, there are two main drawbacks of this similarity metric. Firstly, the GP of the training task may yield predictions with high uncertainty at observations of the target task, which can be caused by the distribution of the training dataset. Secondly, the fidelity of the similarity measure in Equation 4 is reduced when there are few comparison data points, as exemplified by the scenario where the size of $\mathcal{D}_t$ is small. This results in false similarity scores between two tasks during the initial iterations of the BO method. To mitigate the above disadvantages, we propose a novel ranking-based measurement. Specifically, at each iteration t and training task k, we define:

$$\mathcal{D}_t^{(k)} = \{(x, y) \in \mathcal{D}^{(k)} \mid \sigma_t(x) < \epsilon\} \tag{5}$$

where $\epsilon > 0$ is the pre-defined hyper-parameter and $\sigma_t(.)$ is the standard deviation of target GP at iteration t. The proposed similarity score is defined as:

$$S\left(f^{(k)}, f|\mathcal{D}_t^{(k)}\right) = \begin{cases} \frac{\Sigma_{x_i, x_j \in \mathcal{D}_t^{(k)}} \mathbb{I}[(\mu_t(x_i) < \mu_t(x_j)) \otimes (y_i < y_j)]}{|\mathcal{D}_t^{(k)}|\left(|\mathcal{D}_t^{(k)}|-1\right)} & \text{if } |\mathcal{D}_t^{(k)}| > m \\ 0 & \text{else} \end{cases} \tag{6}$$

where $\mu_t(.)$ is the mean of target GP at iteration t; $\mid \mathcal{D}_t^{(k)} \mid$ denotes the number of data points in $\mathcal{D}_t^{(k)}$; and $m > 2$ is the pre-defined threshold. By limiting the variance below the threshold ϵ, the target GP will predict each point in $\mathcal{D}_t^{(k)}$ with high certainty. Moreover, if ϵ is set not too small, the size of $\mathcal{D}_t^{(k)}$ can be large even in the first few iterations, enhancing the reliability of the similarity score. Note that $S\left(f^{(k)}, f \mid \mathcal{D}_t^{(k)}\right)$ will return zero if $\mid \mathcal{D}_t^{(k)} \mid$ less than m. In this case, most of the data points in the training task exhibit high variance under the target GP. This circumstance may arise when the observations of the target task deviate significantly from the distribution of the training task. Consequently, the training task is considered unreliable, leading to a similarity score of zero.

Another advantage of Equation 6 compared to Equation 4 is the computational complexity. The computational complexity when using Equation 4 at iteration t for calculating the similarity score is $\mathcal{O}(t^2Kn^3)$, where K is the number of training tasks and $n = \max_{k\in[K]} n_k$ is the maximum number of observations in training tasks. On the other hand, the computational complexity of Equation 6 at iteration t is $\mathcal{O}(n^2Kt^3)$, which is much lower since $n >> t$.

Determining $\hat{\mathcal{X}}_t'$ Based on the similarity measurement, in each iteration t, we calculate the similarity $S\left(f^{(k)}, f|\mathcal{D}_t^{(k)}\right)$ of the test task f and every training task $f^{(k)}$, where $k \in [K]$. Denote the point with the highest evaluation of the training task $f^{(k)}$ by $x^{*,k}$. We define

$$c_t^{(1)} = \Sigma_{k=1}^K w_t^{(k)} x^{*,k}, \tag{7}$$

where the weight of $x^{*,k}$ is defined as $w_t^{(k)} = \frac{S\left(f^{(k)}, f|\mathcal{D}_t^{(k)}\right)}{\Sigma_{k=1}^{K} S\left(f^{(k)}, f|\mathcal{D}_t^{(k)}\right)}$. The point $c_t^{(1)}$ characterizes the promising position balancing among all the training tasks. The higher similarity of f and $f^{(k)}$ implies the higher weight $w_t^{(k)}$, and hence $c_t^{(1)}$ is closer to the best evaluation point $x^{*,k}$ of the training task $f^{(k)}$.

When the target task is quite different from all source tasks, we have less confidence that the optimum of the test task is close to the best evaluations of the training tasks, so we need to explore other regions than the ones surrounding the best evaluations of the training tasks. Such a promising region is potentially surrounding the best solution found among the evaluation points $\{x_1, ..., x_t\}$ of the test task f because this region has a higher probability of finding a new solution improving over the current solutions. We denote this solution by $c_t^{(2)}$. Next, we define

$$c'_t = \alpha_t c_t^{(1)} + (1 - \alpha_t) c_t^{(2)}, \tag{8}$$

where $c_t^{(1)}$ is determined by Equation 7; $c_t^{(2)}$ is the best solution found of $f(x)$ up to iteration t; and $0 \leq \alpha_t \leq 1$ is the trade-off coefficient at iteration t. The point c'_t is the position balancing between $c_t^{(1)}$ which characterizes the promising region generated by offline data of training tasks, and $c_t^{(2)}$ which characterizes the promising region generated from online data of the test task.

The box $\hat{\mathcal{X}}'_t$ is constructed from $\hat{\mathcal{X}}_{t-1}$ by shifting the center of $\hat{\mathcal{X}}_{t-1}$, denoted by $\mathbf{c}_{t-1}$ toward c'_t, while retaining the size of $\hat{\mathcal{X}}_{t-1}$. The region $\hat{\mathcal{X}}'_t$ is a balance between the promising region generated by the best evaluation points of training tasks which are most similar to the test task, and the promising region generated by the best evaluation point of the test task. In our experiments, we choose $\alpha_t = \frac{\Sigma_{k\in[K]} S\left(f^{(k)}, f|\mathcal{D}_t^{(k)}\right)}{K}$. When α_t is large, the region $\hat{\mathcal{X}}'_t$ has a tendency to move to regions of training tasks where we have confidence that the optimum of f belongs to with high probability. Otherwise, $\hat{\mathcal{X}}'_t$ has a tendency to move to the region surrounding the best solution so far of f. However, since the center c'_t of $\hat{\mathcal{X}}'_t$ may translate fast, which could cause the divergence, we use a fixed domain, denoted by $\overline{\mathcal{X}}_0$ to restrict the translation of c'_t. This domain is the minimum box bounding all the best evaluation points of previous tasks. We translate c'_t toward a point c''_t such that $c''_t \in \overline{\mathcal{X}}_0$ and is the closest point to c'_t. In conclusion, c''_t is the final center of bounding box $\hat{\mathcal{X}}'_t$, which can balance between transfer knowledge and target observations.

5.2 Expanding Strategy

At first glance, one might consider progressively widening the search space at a substantial rate to guarantee the containment of the global optimum of the target task. However, an aggressive expansion rate would result in rapid growth of the search space volume, leading to an increased exploration phase. Consequently, the BO method would fail to achieve sub-linear regret. To solve this issue, we use a novel expanding strategy. Particularly, the size of bounding box $\hat{\mathcal{X}}_t$ is

expanded from $\hat{\mathcal{X}}'_{t-1}$ by the $\left(\frac{b_0^{(i)}-a_0^{(i)}}{t}\right)$ increment in each direction $1 \leq i \leq d$. More precisely, $a_t^{(i)} = \overline{a_t^{(i)}} - \frac{b_0^{(i)}-a_0^{(i)}}{2t}$ and $b_t^{(i)} = \overline{b_t^{(i)}} + \frac{b_0^{(i)}-a_0^{(i)}}{2t}$, where $\overline{a_t^{(i)}}, \overline{b_t^{(i)}}$ are the lower and upper bound at dimension i of $\hat{\mathcal{X}}'_t$, respectively. The expansion increment $\mathcal{O}\left(\frac{1}{t}\right)$ is small enough so that the search space is expanded slowly as the iteration t tends to ∞ but we still ensure that the optimum will be found. Importantly, we will show that this expanding strategy combined with the moving strategy in section 5.1 can achieve the sub-linear regret in the below section.

5.3 Regret Analysis

We have the following theorem.

Theorem 1. *Let $f \sim \mathcal{GP}(\mathbf{0}, k)$ with a stationary covariance function k. Assume that there exist constants $s_1, s_2 > 0$ such that $\mathbb{P}[sup_{\mathbf{x}\in\mathcal{X}}|\partial f/\partial x_i| > L] \leq s_1 e^{-(L/s_2)^2}$ for all $L > 0$ and for all $i \in \{1, 2, ..., d\}$. Pick a $\delta \in (0, 1)$. Set $\beta_T = 2log(4\pi_t/\delta) + 4dlog(dTs_2(1 + ln(T))\sqrt{log(4ds_1/\delta)})$. There is a constant $T_0 > 0$ which is independent of t such that for any horizon $T > T_0$, the cumulative regret of the proposed global BO algorithm (Algorithm 1) is bounded as*

$$R_T \leq \sum_{t=1}^{T_0} L||x^* - c_t''||_1 + \sqrt{C_1 T \beta_T \gamma_T(\mathcal{C}_T)} + \frac{\pi^2}{6}$$

, with probability $1-\delta$, where the box $\mathcal{C}_T$ is the box covering $\overline{\mathcal{X}}_0 = \prod_{i=1}^{d}[a^{(i)}, b^{(i)}]$ and $\hat{\mathcal{X}}_t$. It is computed as

$$\mathcal{C}_T = \prod_{i=1}^{d}\left[a^{(i)} - \frac{(b_0^{(i)} - a_0^{(i)})}{2}\sum_{j=1}^{T}\frac{1}{j}, b^{(i)} + \frac{(b_0^{(i)} - a_0^{(i)})}{2}\sum_{j=1}^{T}\frac{1}{j}\right]$$

, and $\gamma_T(\mathcal{C}_T)$ is the maximum information gain for any T observations in the domain $\mathcal{C}_T$ (see [21]). This term is computed as follows:

- *For SE kernels: $\gamma_T(\mathcal{C}_T) = \mathcal{O}((ln(T))^{d+1})$,*
- *For Matérn kernels with $\nu > 1$: $\gamma_T(\mathcal{C}_T) = \mathcal{O}(T^{\frac{d(1+d)}{2\nu+d(d+1)}})$*

Compared to the regret bound of the traditional BO with a fixed search space and without transfer learning, our regret bound has additional components: $\sum_{t=1}^{T_0} L||x^* - c_t''||_1$ and the $\gamma_T(\mathcal{C}_T)$. Since T_0 is a constant; L is a constant and $||x^* - c_t''||_1$ is bounded by the diameter of boxes $\mathcal{C}_t$ with $t \leq T_0$, this component $\sum_{t=1}^{T_0} L||x^* - c_t''||_1$ is a constant which is independent of t. In addition, although the search space is expanded over iterations, the maximum information gain of these expanded search spaces $\gamma_T(\mathcal{C}_T)$ is still bounded by $\mathcal{O}((\ln(T))^{d+1})$ for SE kernels and by $\mathcal{O}(T^{\frac{d^2(1+d}{2\nu+d(d+1)}})$ with Matérn kernels. Interestingly, we remark that these bounds of $\gamma_T(\mathcal{C}_T)$ have the same order as the ones for BO with a

fixed search space (see Theorem 5 of [21]). As a result, the regret bound of our proposed algorithm is sub-linear in T for SE kernels and Matérn kernels (with several conditions on d and ν).

In our regret bound, the point c_t'' reflects c_t' which is defined in Section 5.1. It represents the center of promising regions where we have confidence that the optimum x^* belongs to with high probability. Recall that c_t' is the position balancing between $c_t^{(1)}$ which characterizes the promising region generated by the best evaluations of training tasks, and $c_t^{(2)}$ which is the best solution up to iteration t of the test task. Therefore, if c_t' is closer to x^* then c_t'' is closer to x^* and, hence the regret bound is tighter. In our experiments, we see that c_t' moves close to x^* quite quickly (see Figure 5). Moreover, the constant T_0 can be reduced with the help of transfer knowledge as illustrated in section 6.4. **A full proof of Theorem 1 is provided in Appendix A**.

6 Experiments

In this section, we demonstrate the effectiveness of our proposed methods across a wide range of black-box functions. We compare our method with seven other benchmark methods: **GP-based BO**; **Box-BO** [17], which designs a search space using a box bounding the best evaluations of training tasks; **Ellipsoid-BO** [17], which designs a search space using a low-volume ellipsoid bounding the best evaluations of training tasks; **US-BO** (**U**ncertainty **S**earch space Bayesian Optimization) [12], which learns search spaces by using the similarity between tasks and a GP classifier; **UBO** [9], which expands the search space whenever the local $\epsilon-$accuracy condition is satisfied; **FBO** [16], which broadens the search space using a filtering expansion strategy; and a method for high-dimensional BO **TuRBO** [5], which uses a trust region centered at the best solution. Note that both **UBO** and **FBO** are designed for an unbounded search space without leveraging transfer knowledge. To underscore the effectiveness of the proposed similarity score, we also compare our method with two variants: i) **Old-sim**, employing the similarity in Equation 4; and ii) **No-transfer**, wherein the center is moved solely to the best-observed point so far. The performance of the methods is quantified using log regret, and each experiment is repeated 20 times. To ensure fairness in our experiments, we adopt a uniform sampling approach, selecting three initialization points within the low-volume bounding box recommended by Box-BO. This initialization strategy aims to introduce additional information from training tasks to non-transfer methods, such as TuRBO, FBO, UBO and No-transfer. For our method, we set $\epsilon = \sqrt{\frac{k(0,0)}{2}}$ and $m = 2d$, where $k(.,.)$ is the GP kernel and d is the input dimension. Moreover, we set the initial region $\hat{\mathcal{X}}_0$ as 20% of the restricted domain $\overline{X}_0$ stated in section 5.1. **Due to space limitation, we also provide the empirical analysis of the proposed method's hyper-parameters, e.g ϵ, m, proportion of initial region in Appendix C to show the robustness of our method**.

The code is available at https://github.com/Fsoft-AIC/BO-transfer-search-space

6.1 Experiments on synthetic functions

We evaluate our method and the benchmark methods on several types of standard optimization benchmark functions. We design the synthetic experiments in three scenarios with different input dimensions: **Scenario 1** *the global optimum of the target task is in proximity to the best evaluations of all training tasks*; **Scenario 2** *the global optimum of the target task is distant from all the best evaluations of training tasks*; **Scenario 3** *the global optimum of the target task is close to some of the best evaluations of training tasks and distant from others.* For the first scenario, we selected Ackley ($d = 4$), Powell ($d = 4$), Dixon-Price ($d = 5$), and Levy ($d = 8$) as the objective functions. To construct the training datasets, we apply the random translations and rescalings of up to +/- 30% to the x and y values, respectively. For the second scenario, we test the algorithms on four benchmark functions: Styblinski-Tang ($d = 4$), Ackley ($d = 5$), Rosenbrock ($d = 6$) and Griewank ($d = 30$). Different from the first scenario, we applied translation for x as well as scalings for y up to +/- 50% to create the training datasets, thereby decreasing the similarity between the training tasks and the target task. Additionally, to make the experiments more challenging, we shifted the search space of each training dataset so that it does not contain the true optimum of the target function. Consequently, the optimum point of the target task becomes far from the optimum of the training tasks. Lastly, the third scenario is a mixture of the first scenario and the second scenario. We select Hyper-Ellipsoid ($d = 5$), Ackley ($d = 6$), Rastrigin ($d = 10$) and Perm ($d = 20$) as the target tasks. To generate the training tasks, we follow a similar procedure as described in the first scenario to create m_1 tasks. Additionally, we employ the mechanism outlined in the second scenario to construct m_2 training tasks. For each objective function, we created 15 training datasets, each consisting of 1500 data points. For scenario 3, we choose $m_1 = 10$ and $m_2 = 5$. The experimental results are reported in Figure 2, 3, 4.

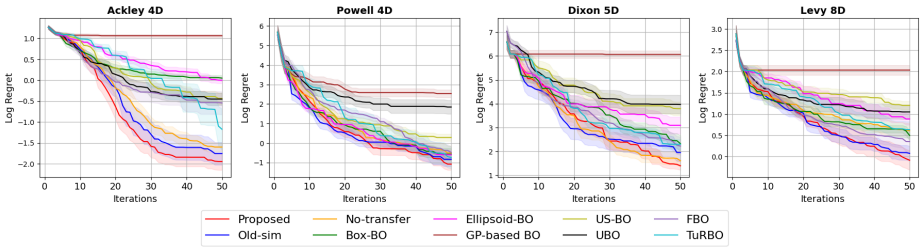

Fig. 2. Performances on four standard functions in Scenario 1. The $y-$axis presents the log regret (a smaller value is better).

We consider the first scenario where the global optimum of the target task is close to the best evaluation of the training tasks. Although other transfer learning-based methods like Box-BO, Ellipsoid-BO, and US-BO are also designed

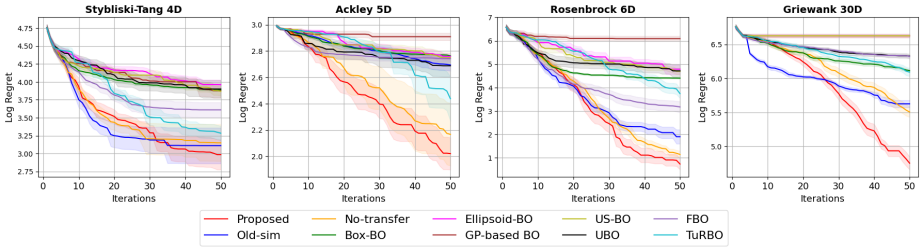

Fig. 3. Performances on four standard functions in Scenario 2. The $y-$axis presents the log regret (a smaller value is better).

for this scenario, our method outperforms the other benchmark methods in all test functions. This superiority stems from its ability to dynamically adapt and shift towards promising regions as illustrated in Figure 5 (Left). US-BO performs poorly in experiments due to the challenges in locating the optimal point of the acquisition in their extracted region. Turbo uses an adaptive shifting strategy but does not leverage the knowledge of training tasks, and hence performs poorly compared to our methods. Notably, the Old-sim method yields competitive outcomes with our proposed approach. This can be attributed to the high similarity between the training tasks and the target task, indicating that even with limited comparison points, the similarity metric defined in Equation 4 remains reliable for each training task.

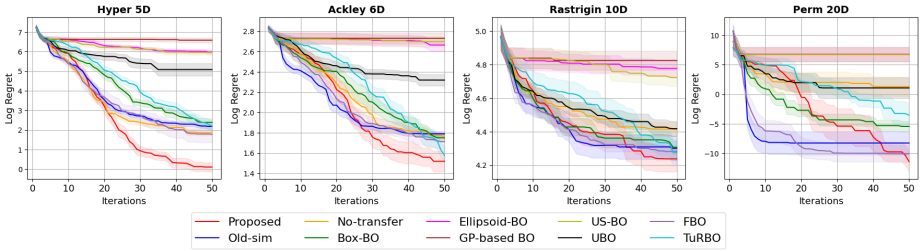

Fig. 4. Performances on four standard functions in Scenario 3. The $y-$axis presents the log regret (a smaller value is better).

For the second scenario where the global optimum of the target task is further to the best evaluations of the training tasks, our proposed method shows the best results compared to the baselines (Figure 3). In this setting, the resemblance between the training tasks and the test tasks diminishes. However, our methods are adaptive in the sense that they can move to regions surrounding the best solution so far of the test task rather than regions containing evaluations of the training tasks if the similarity score is low. We illustrate this movement in Figures 5 (Right) as an example. As can be seen in Figure 3, the other transfer

learning-based search space designing methods struggle, as they primarily concentrate on the local region around the best evaluation points of the training tasks. Turbo shows a good performance due to the movement of the trust region. In contrast with scenario 1, the performance of Old-sim is reduced since the similarity in Equation 4 is unreliable with few comparison points for dissimilar training tasks, which can lead to a falsely high similarity score. Conversely, the No-transfer method exhibits competitive performance by disregarding information from dissimilar training tasks. Similar to Scenario 1, our method consistently outperforms the compared counterparts, underscoring the effectiveness of the proposed similarity score. Moreover, even for the high-dimensional Griewank function, our method performs more efficiently than TurBO.

A comparable trend is observed in Scenario 3, as depicted in Figure 4. Our proposed method demonstrates the most favorable outcome, maintaining its effectiveness even as the input dimension is increased for the Perm test function. This outcome underscores the capability of the proposed similarity metric to identify a substantial portion of the training dataset that exhibits strong similarity to the target function, even within a mixed setting.

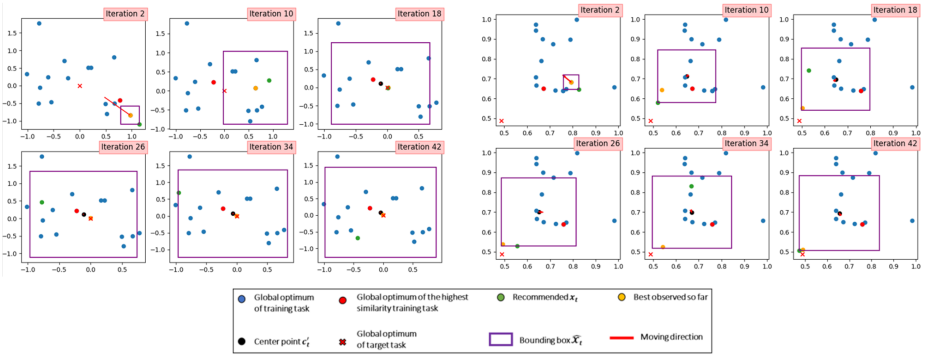

Fig. 5. (**Left**) Progress of proposed method on Ackley 4D function. In this scenario, the optimum point of the target task is close to the best evaluation of training tasks; (**Right**) Progress of proposed global method on Rosenbrock 6D function. In this scenario, the optimum point of the target task is far away from the best evaluation of training tasks. A full progress in given in Appendix B.

6.2 Experiments on hyperparameter tuning

In this part, we assess the effectiveness of our method in deep learning algorithm tuning problems. Specifically, we utilize the ResNet Tuning Benchmark introduced by [12], which involves optimizing five hyperparameters of ResNet on datasets such as CIFAR-10, SVHN, and Tiny-Imagenet. Due to space limitation, we added an experiment about tuning hyper-parameters of SGD for ridge

regression in Appendix D. Differing from [12], we focus on datasets where the hyperparameter *nesterov* is set to *False*, while optimizing the remaining four hyperparameters. Our evaluation is based on the Normalized Classification Error (NCE) [12] as depicted in Figure 6. It is important to note that the experiments are conducted with discrete input domains, and the search space configuration is predefined. Since the input domains are discrete, we set initial region $\hat{\mathcal{X}}_0$ larger with the rate of 80% of the restricted domain $\overline{X}_0$. Overall, our proposed method consistently outperforms the compared baseline methods. Additionally, we observed that moving-based strategies, such as TuRBO, No-transfer, and Old-sim, demonstrate commendable performance even within the predefined search space. Conversely, UBO and FBO methods exhibit poor performance in the context of discrete input domains.

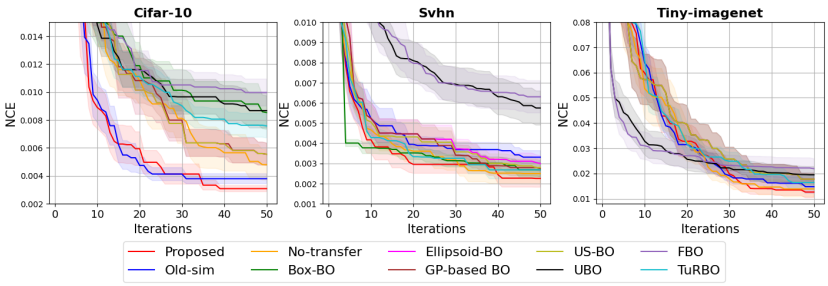

Fig. 6. Performances on tuning ResNet for three vision problems. The $y-$axis presents the NCE (a smaller value is better).

6.3 Experiments on more real-world applications

To further demonstrate the performance of our methods in real-world applications, we consider three real-world tasks including location selection for oil wells ($d = 4$) [10]; robot pushing problem ($d = 14$) [5], and rover trajectory planning problem ($d = 20$) [5]. We assume that the search space of all three real-world applications is unknown.

Oil Well problem. The objective of this task is to determine the deepest drilled depth among oil wells based on the longitude and latitude coordinates of both the surface and bottom of the well. Following a methodology similar to [10], we utilize 30 datasets, each comprising over 1000 parameter configurations. We evaluate the transfer learning capabilities using a leave-one-task-out approach, wherein one dataset is reserved for testing while the remaining datasets serve as training tasks. It is worth noting that the input domains for this experiment are discrete. For evaluation, we employ the Normalized Classification Error metric [12]. The experimental results are depicted in Figure 7 (Left), indicating that our methods outperform the baseline approaches. The relatively lower performance

of TuRBO can be attributed to the following reasons: given the discrete input dimension of this benchmark, the size of the trust region becomes too small, resulting in a limited number of candidates available for evaluation within that region. In contrast, our methods progressively expand the search space, thereby circumventing this issue.

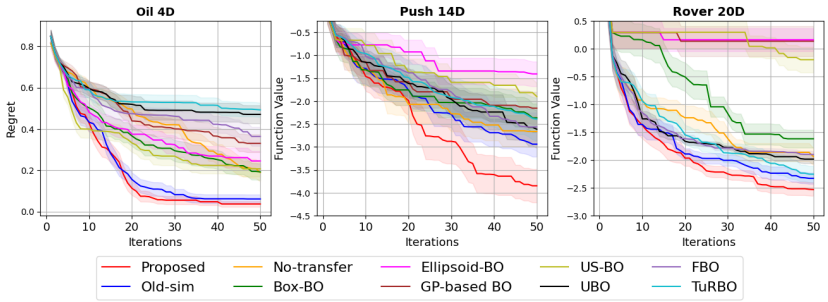

Fig. 7. Performances on three real-world applications. The $y-$axis presents the regret and the function value (a smaller value is better).

Robot pushing problem. The goal of this task is to control 14 parameters of robot hands to push two objects toward the designated goal location [5]. To create the training task, we uniformly sample the first dimension and the second dimension of initial positions in a range $(-0.5, 0.5)$ and $(-2, 2)$, respectively. We use the default initial location stated in [5], which is $(0, 2)$ and $(0, -2)$ for the target task. The goal location for all datasets is the same. We created 15 training datasets with different initial locations, each consisting of 2000 data points. The function values are reported in Figure 7 (Middle). Our method outperforms all other methods after a few iterations, consistently demonstrating the best performance. In contrast, US-BO and Ellipsoid-BO exhibit the poorest results, partially attributed to challenges in optimizing the acquisition within their designated spaces. The No-transfer and Old-sim methods yield competitive results, underscoring the effectiveness of the moving and scaling approach. On the other hand, FBO, UBO, and TuRBO display similar performances.

Rover trajectory planning problem. The goal of this task is to optimize the locations of 10 points in the $2D$ plane that determine the trajectory of a rover via BSpline method [5]. Like the robot pushing problem, we uniformly sample the start position in a box $[0, 1]^2$, while the start position of the target task is $(0.05, 0.05)$. The goal position for all tasks is the same. Overall, we had 15 training datasets each consisting of 2000 data points. The outcome is illustrated in Figure 7 (Right), where our method exhibits superior performance compared to the alternatives. US-BO and Ellipsoid-BO encounter scalability issues as the

input dimension increases. TuRBO performs well in high-dimensional inputs, while Box-BO appears to be constrained within its extracted region.

6.4 Experiments on the search space design

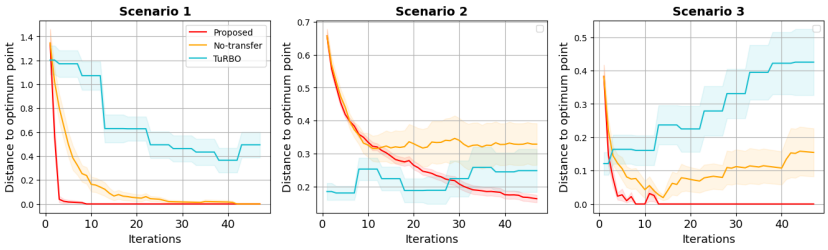

Fig. 8. Distance to the optimum point on three scenarios. The $y-$axis presents value in Equation 9 (a smaller value is better).

In this subsection, we study the ability to contain the global optimum of the target task of our methods on the three scenarios stated in 6.1. For each method, we report the distance from the global optimum point of the target task to their search spaces at every iteration:

$$d_t(M) = \left|\left|x^* - \text{argmin}_{x \in B_t(M)} ||x^* - x||_2\right|\right|_2, \tag{9}$$

where $d_t(M)$ is the distance from the global optimum point of the target task to the extracted search space of method M in iteration t, x^* is the global optimum of the target task, $B_t(M)$ is the extracted region of method M in iteration t and $||.||_2$ is the l_2 norm. From Equation 9, the search space contains the global optimum of the target task if $d_t(M) = 0$. We conduct a comparison of our methods with TuRBO and the No-transfer methods across different scenarios: Powell 4D in Scenario 1, Rosenbrock 6D in Scenario 2, and Hyper-Ellipsoid 5D in Scenario 3. The results are presented in Figure 8. In Scenario 1 (Figure 8 Left), our method rapidly converges towards the global optimum of the target task, outperforming the No-transfer method, which attains containment of x^* at a later stage. This highlights the effectiveness of transferring knowledge. TuRBO faces challenges as its trust region fails to contain x^* due to fluctuations in $d_t(.)$ during the process. In Scenario 2 (Figure 8 Middle), where $x*$ is distant from the global optimum of the training tasks, our proposed method significantly reduces the gap to the optimum point at a faster rate than the No-transfer method. TuRBO exhibits a similar performance as in Scenario 1. In Scenario 3 (Figure 8 Right), the proposed method successfully reaches the optimum point within 20 iterations, while TuRBO diverges away from x^*. In summary, when the training tasks exhibit strong similarity, our method's bounding box can rapidly

encapsulate x^* within the initial iterations, resulting in a small value for the term T_0 in Theorem 1. Further experiments to show the balancing between distance to the global optimum and search space area can be found in Appendix E.

Acknowledgments. Hung The Tran and Dung D. Le acknowledge the support of the Center for Environmental Intelligence at VinUniversity (project VUNI.CEI.FS_0007).

References

1. Bai, T., Li, Y., Shen, Y., Zhang, X., Zhang, W., Cui, B.: Transfer learning for bayesian optimization: A survey. arXiv preprint arXiv:2302.05927 (2023)
2. Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. Advances in neural information processing systems **24** (2011)
3. Chen, W., Fuge, M.: Adaptive expansion bayesian optimization for unbounded global optimization (2020)
4. Dai, Z., Chen, Y., Yu, H., Low, B.K.H., Jaillet, P.: On provably robust meta-bayesian optimization. In: Uncertainty in Artificial Intelligence. pp. 475–485. PMLR (2022)
5. Eriksson, D., Pearce, M., Gardner, J.R., Turner, R., Poloczek, M.: Scalable Global Optimization via Local Bayesian Optimization. Curran Associates Inc., Red Hook, NY, USA (2019)
6. Fan, Z., Han, X., Wang, Z.: Transfer learning for bayesian optimization on heterogeneous search spaces. Transactions on Machine Learning Research (2024)
7. Feurer, M.: Scalable meta-learning for bayesian optimization using ranking-weighted gaussian process ensembles (2018)
8. Feurer, M., Letham, B., Hutter, F., Bakshy, E.: Practical transfer learning for bayesian optimization (2018)
9. Ha, H., Rana, S., Gupta, S., Nguyen, T.T., Tran-The, H., Venkatesh, S.: Bayesian optimization with unknown search space. In: NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada. pp. 11772–11781 (2019)
10. Hsieh, B.J., Hsieh, P.C., Liu, X.: Reinforced few-shot acquisition function learning for bayesian optimization. Advances in Neural Information Processing Systems **34**, 7718–7731 (2021)
11. Kandasamy, K., Schneider, J., Póczos, B.: High dimensional Bayesian optimisation and bandits via additive models. In: International conference on machine learning. pp. 295–304. PMLR (2015)
12. Li, Y., Shen, Y., Jiang, H., Bai, T., Zhang, W., Zhang, C., Cui, B.: Transfer learning based search space design for hyperparameter tuning. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. p. 967–977. KDD '22, Association for Computing Machinery, New York, NY, USA (2022)
13. Li, Y., Shen, Y., Jiang, H., Zhang, W., Yang, Z., Zhang, C., Cui, B.: ACM (aug 2022)
14. Marco, A., Berkenkamp, F., Hennig, P., Schoellig, A.P., Krause, A., Schaal, S., Trimpe, S.: Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 1557–1563. IEEE (2017)
15. Mockus, J.: On bayesian methods for seeking the extremum. In: Proceedings of the IFIP Technical Conference. pp. 400–404. Springer-Verlag, London, UK, UK (1974)

16. Nguyen, V., Gupta, S., Rane, S., Li, C., Venkatesh, S.: Bayesian optimization in weakly specified search space. In: 2017 IEEE International Conference on Data Mining (ICDM). pp. 347–356 (2017)
17. Perrone, V., Shen, H.: Learning search spaces for bayesian optimization: Another view of hyperparameter transfer learning. In: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada.
18. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press (2005)
19. Shahriari, B., Bouchard-Cote, A., Freitas, N.: Unbounded bayesian optimization via regularization. In: Gretton, A., Robert, C.C. (eds.) Proceedings of the 19th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 51, pp. 1168–1176. PMLR, Cadiz, Spain (09–11 May 2016)
20. Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M.M.A., Prabhat, P., Adams, R.P.: Scalable bayesian optimization using deep neural networks. In: ICML 2015 - Volume 37. p. 2171–2180.
21. Srinivas, N., Krause, A., Kakade, S.M., Seeger, M.W.: Information-theoretic regret bounds for gaussian process optimization in the bandit setting. IEEE Trans. Inf. Theor. **58**(5), 3250–3265 (May 2012)
22. Swersky, K., Snoek, J., Adams, R.P.: Multi-task bayesian optimization. In: Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. (eds.) NeuRIPS. vol. 26 (2013)
23. Tran-The, H., Gupta, S., Rana, S., Ha, H., and Venkatesh, S.: Sub-linear regret bounds for bayesian optimisation in unknown search spaces, Advances in Neural Information Processing Systems, volume 33, pages 16271–16281. (2020)
24. Tran-The, H., Gupta, S., Rana, S., and Venkatesh, S.: Regret bounds for expected improvement algorithms in gaussian process bandit optimization, Proceedings of The 25th International Conference on Artificial Intelligence and Statistics, volume 151 of Proceedings of Machine Learning Research, pages 8715–8737. PMLR (2022)
25. Tran-The, H., Gupta, S., Rana, S., and Venkatesh, S.: Trading convergence rate with computational budget in high dimensional bayesian optimization. Proceedings of the AAAI Conference on Artificial Intelligence, 34(03):2425–2432. (2022)
26. Volpp, M., Fröhlich, L.P., Fischer, K., Doerr, A., Falkner, S., Hutter, F., Daniel, C.: Meta-learning acquisition functions for transfer learning in bayesian optimization. In: International Conference on Learning Representations (2020)
27. Wistuba, M., Schilling, N., Schmidt-Thieme, L.: Scalable gaussian process-based transfer surrogates for hyperparameter optimization. Machine Learning **107**, 43–78 (2017)
28. Yogatama, D., Mann, G.S.: Efficient transfer learning method for automatic hyperparameter tuning. In: International Conference on Artificial Intelligence and Statistics (2014)
29. Nguyen, Minh-Duc and Dinh, Phuong Mai and Nguyen, Quang-Huy and Hoang, Long P. and Le, Dung D., "Improving Pareto Set Learning for Expensive Multi-objective Optimization via Stein Variational Hypernetworks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 18, pp. 19677–19685, Apr. 2025.
30. Nguyen, Quang-Huy and Hoang, Long P. and Viet, Hoang V. and Le, Dung D., "Controllable Expensive Multi-objective Learning with Warm-starting Bayesian Optimization," arXiv preprint, arXiv:2311.15297, 2024.