Proceedings of Machine Learning Research vol 278:1–16, 2025 2025 2nd International Conference on Machine Learning and Intelligent Computing

Trust-Region Bayesian Optimization for High-Dimensional Black-Box Problems: Integrating Deep Kernel Learning with Adaptive Gradient Mechanisms

Guolin Yang 2213393058@ST.GXU.EDU.CN
School of Computer, Electronics and Information, Guangxi University, Nanning 53004, China

Hua Qin[*] QINHUA263@163.COM
School of Computer, Electronics and Information, Guangxi University, Nanning 53004, China

Yanming Fu FYM2005@126.COM
School of Computer, Electronics and Information, Guangxi University, Nanning 53004, China
[*]*Corresponding author*

Editors: Nianyin Zeng, Ram Bilas Pachori and Dongshu Wang

Abstract

The traditional Bayesian optimization (BO) algorithm faces significant performance bottlenecks when addressing high-dimensional black-box optimization problems. To mitigate this challenge, the present paper introduces a novel trust region Bayesian optimization algorithm. Firstly, in the design of the BO surrogate model, we employ a combination of deep neural networks and kernel methods to enhance the Gaussian process regression (GPR) model. This approach improves GPR's capacity to identify and fit the nonlinear characteristics of black-box functions while also increasing regression accuracy. Secondly, in formulating the BO acquisition function, an adaptive gradient trust region adjustment method is utilized to bolster BO's search capabilities within high-dimensional solution spaces. Concurrently, a hybrid sampling strategy is implemented to generate more diverse sampling points, thereby enhancing BO's ability to escape local optima. The proposed algorithm has been validated on three 60D multimodal complex functions as well as two engineering application problems and compared with other advanced variants of BO. Experimental results demonstrate that our proposed algorithm exhibits superior iterative convergence, rapidly approaches optimal values for black-box problems with fewer function evaluations, and achieves higher computational accuracy. These findings confirm both the feasibility and effectiveness of the improved BO approach presented in this paper.

Keywords: High-dimensional black-box optimization problems; Bayesian optimization; Deep kernel learning; Adaptive gradient; Hybrid sampling strategy.

1. Introduction

There exist numerous high-dimensional black-box optimization problems in the realms of scientific research and engineering practice (Greenhill et al., 2020). These challenges are characterized by a multitude of decision variables, with their true mathematical models often remaining elusive. Typically, only a limited set of input-output sample data from the black box can be utilized to extract pertinent information regarding the underlying function. BO offers a framework for modeling these black-box optimization issues through surrogate models and available sample data, thereby finding extensive applications in various domains such as hyperparameter tuning (Yu and Zhu, 2020), material selection (Yamamoto et al., 2025), and drug design (Bunn et al., 2023). The most commonly employed surrogate model within Bayesian optimization is GPR. This approach updates the posterior distribution based on the input-output sample data derived from the black-box function and subsequently identifies the next evaluation point via an acquisition function. The objective is to approximate the optimal solution of the original problem while minimizing evaluations of the black-box function. However, traditional Bayesian optimization techniques encounter significant challenges when addressing high-dimensional black-box optimization problems. For instance, once problem dimensions exceed 20, GPR's computational complexity escalates markedly with increasing dimensionality; additionally, capturing both local and global characteristics of intricate black-box functions becomes increasingly difficult. This often results in suboptimal solution quality (Frazier, 2018). The primary challenges that BO faces in addressing high-dimensional problems can be categorized into three main factors: Firstly, Brochu Eric et al. (2007) identified that when tackling high-dimensional issues, BO is susceptible to becoming trapped in local optima, which significantly hampers the subsequent optimization process and leads to near stagnation. Secondly, the objective function of the original problem is typically nonlinear. Given a limited number of observed samples, GPR struggles to accurately capture the characteristics of the black-box function. Lastly, traditional sampling strategies employed by BO acquisition functions—such as probability of improvement (PI) and expected improvement (EI)—exhibit a "short-sighted phenomenon." This results in an imbalance between exploration and exploitation during the search for optimal solutions within the solution space.

To enhance the performance of Bayesian Optimization (BO) in addressing high-dimensional optimization challenges, various studies have proposed strategies to improve BO's modeling capabilities for high-dimensional black-box problems by either replacing or augmenting the Gaussian Process Regression (GPR) model. For example, Hutter et al. (2011) substituted the GPR model with a random forest and introduced the Tree-structured Parzen Estimator (TPE) algorithm; McIntire et al. (2016) utilized a sparse Gaussian process as a probabilistic surrogate model, leading to the development of the Sparse Gaussian Process Bayesian Optimization (SGP-BO) algorithm; Wang et al. (2018) presented Ensemble Bayesian Optimization (EBO), which is based on generating a batch of diverse sampling points. Eriksson et al. (2019) introduced a local Bayesian optimization algorithm known as trust region Bayesian optimization (TuRBO), which employs a trust region strategy that demonstrates superior search capabilities within the solution space when compared to the sampling strategies utilized in traditional Bayesian optimization. Consequently, the optimization performance of this algorithm significantly surpasses that of conventional approaches. Wilson et al. (2016) proposed a method termed deep kernel learning (DKL), which integrates deep learning architectures with kernel methods to enhance the modeling capacity and scalability of Bayesian optimization for black-box functions. Inspired by the aforementioned literature, this paper proposes a deep kernel learning gradient adaptive trust region Bayesian optimization (DKG-TBO) to further enhance the performance of BO in addressing high-dimensional black-box optimization problems. The proposed algorithm integrates the feature extraction capabilities of deep neural networks with the nonlinear identification strengths of kernel methods, thereby improving the modeling performance of GPR for black-box functions. It dynamically adjusts the trust region radius based on gradient information to increase search efficiency within the solution space of high-dimensional challenges. Additionally, it employs a hybrid sampling strategy to generate more diverse evaluation points, enhancing BO algorithms' ability to escape local optima. Finally, we validate the proposed algorithm on three nonlinear multimodal test functions and two engineering application problems while comparing its performance against classical BO algorithms. Experimental results demonstrate that DKG-TBO exhibits significant advantages in solving high-dimensional black-box optimization problems.

2. Related work

2.1. Bayesian optimization algorithm

An unconstrained minimization problem can be articulated as follows:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{X}\subseteq\mathbb{R}^d}{\operatorname{argmin}}\ f(\mathbf{x}) \tag{1}$$

Where x is a d dimensional decision vector, and $\mathcal{X}$ represents the decision space. $f(\mathbf{x})$ is the black-box objective function. If the cost of evaluating the black-box function $f(\mathbf{x})$ is high, it implies that the number of evaluations is limited, usually only a few dozen or a few hundred times. For instance, in the problem of developing an effective drug for a certain type of cancer (Wu et al., 2019), the drug formula can be regarded as the decision space, the drug effect as the function output, and clinical trials as the means to evaluate the drug effect. The goal is to find a drug formula that can cure patients with the highest probability. In this problem, the objective function is not an explicit mathematical expression, and the process of evaluating the function may lead to the death of patients. Obviously, such an assessment is costly. In addition, there may be other limitations, such as the fact that each assessment may take a considerable amount of time (usually several hours), or it may be due to the high financial cost of each assessment (for example, purchasing expensive cloud computing resources or laboratory materials).

The principle of Bayesian optimization is to model a black-box function using GPR based on a set of input and output data samples. Subsequently, it generates new evaluation samples through an acquisition function. This iterative process continues in a loop to search for the global optimal solution of the black-box function (Gisperg et al., 2025). The algorithmic framework of a basic Bayesian optimization (BO) algorithm is presented in Table 1.

Table 1: The procedure of the Bayesian optimization algorithm.

```
Input: The objective function $f(.)$, the initial sampling points , and the total number of evaluations
of the objective function$N$.
Output: The optimal solution of the objective function $\mathbf{x}^*$.
1: Determine the GPR surrogate model and select the acquisition function $\alpha(\mathbf{x}; D_n)$.
2: Initialize data samples and evaluate, that is,
$D_{n_0} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{n_0}, y_{n_0})\}$, where $y_i = f(\mathbf{x}_i)$.
3: $n \leftarrow n_0$.
4: while $n \leq N$ do
5: Build a GPR model using the observation set $D_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$.
6: By maximizing the acquisition function $\alpha(\mathbf{x}; D_n)$, the next evaluation point $\mathbf{x}_{n+1}$ is obtained,
that is: $\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} \alpha(\mathbf{x}; D_n)$
7: Evaluate the function value $y_{n+1} \leftarrow f(\mathbf{x}_{n+1})$ of $\mathbf{x}_{n+1}$.
8: Update the observation set $D_{n+1} \leftarrow D_n \cup \{(\mathbf{x}_{n+1}, y_{n+1})\}$.
9: end while
10: Return the optimal solution $\mathbf{x}^*$.
```

2.2. Trust region Bayesian optimization algorithm

The fundamental concept of the trust region algorithm (Yuan, 2000) is to utilize a quadratic model for approximating the black-box function. Initially, an initial point $\mathbf{x}_0$ and an initial trust region radius Δ_0 are selected, along with the establishment of convergence criteria. At the current iteration point $\mathbf{x}_k$, the approximate quadratic model $m_k(d)$ of the black-box function is constructed as follows.

$$\min m_k\left(d\right) = f_k + g_k^T d + \frac{1}{2} d^T B_k d \tag{2}$$

$$s.t. |\mid d \mid|_2 \leq \Delta_k \tag{3}$$

Where f_k, g_k, and B_k denote the objective function value, the gradient, and the approximate Hessian matrix at a given point $\mathbf{x}_k$, respectively. The symbol Δ_k represents the radius of the trust region at $\mathbf{x}_k$.

The trust region is typically represented as a sphere or polyhedron centered around the current optimal solution $\mathbf{x}_k$, with its radius dynamically adjusted in each iteration. When the current approximate model accurately reflects the black-box function, the trust region is expanded, thereby enlarging the search area for potential solutions. Conversely, if the fitting performance is inadequate, the trust region is contracted. The TuRBO algorithm integrates the concept of trust regions with Bayesian optimization, and its procedural steps are outlined as shown in table 2.

Table 2: Framework of the TuRBO algorithm.

Input: objective function $f(.)$, total number of evaluations N, and the initial trust region TR.
Output: The optimal solution of the objective function $\mathbf{x}^*$.

```
1: Initialize the sampling points $n_0$ using the Latin hypercube sampling method (Yang et al., 2023)
and perform black-box function evaluations, that is, $D_{n_0} = \{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^{n_0}$.
2: Determine the GPR surrogate model, $n \leftarrow n_0$.
3: while $n \leq N$ do
4: Conduct GPR modeling on $D_n$.
5: $\mathbf{x}^* = \arg\min_{\mathbf{x}\in D} f(\mathbf{x})$ , and construct a trust region $TR$ centered at $\mathbf{x}^*$ as a hyperrectangle.
6: According to the length scale $\lambda_i$ of each dimension in the GPR model, adjust the radius of
each dimension to maintain the total volume at $L^d$.
7: Determine the next evaluation point $\mathbf{x}_{n+1}$, $\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} \alpha(\mathbf{x}; D_n)$, $y_{n+1} \leftarrow f(\mathbf{x}_{n+1})$.
8:  if consecutive failures, then $L \leftarrow \min\{L_{\max}, 2L\}$. end if
9: if consecutive success, then $L \leftarrow L/2$. end if
10: $D_{n+1} \leftarrow D_n \cup \{(\mathbf{x}_{n+1}, y_{n+1})\}$.
11: end while
12: Return the optimal solution $\mathbf{x}^*$.
```

3. The proposed trust region Bayesian optimization algorithm(DKG-TBO)

3.1. Gaussian Processes for deep kernel learning

When the traditional Bayesian optimization method employs GPR as its surrogate model, it encounters two significant limitations. Firstly, conventional Gaussian processes utilize fixed kernel functions—such as Matern and radial basis function (RBF)—to assess the similarity between different inputs. These kernel functions often struggle to effectively capture the intricate nonlinear structures and potential low-dimensional manifold features present in high-dimensional data. Secondly, when the input dimension exceeds 20, both dimension sensitivity issues and the automatic relevance determination (ARD) mechanism face considerable optimization challenges due to a sharp increase in dimensional parameters.

In this study, we present the deep kernel learning (DKL) method as an extension of the traditional Gaussian process. Our objective is to enhance its feature recognition and modeling capabilities for high-dimensional and complex black-box functions. The original input $\mathbf{x} \in \mathbb{R}^d$ is mapped to a low-dimensional or structured feature space through a parameterized deep neural network denoted as $\phi(\mathbf{x};\theta)$, and is the parameters of the deep neural network. This process is described as follows.

$$\mathbf{z}(\mathbf{x}) = \phi(\mathbf{x};\theta), \quad \mathbf{z}(\mathbf{x}) \in \mathbb{R}^p \tag{4}$$

Subsequently, a kernel function of deep kernel learning in the Hilbert feature space is defined as$k(\mathbf{z}(\mathbf{x}_i), \mathbf{z}(\mathbf{x}_j);\psi)$, and ψ is the hyperparameters of the kernel function. The GPR model is constructed as the following:

$$f(\mathbf{x}) \sim \mathcal{GP}(0, K)) \tag{5}$$

Where K is the kernel matrix generated by mapping the input sample data through a deep neural network, and its elements are given by $K_{ij} = k(\mathbf{z}(\mathbf{x}_i), \mathbf{z}(\mathbf{x}_j);\psi)$.

The final joint training objective is to maximize the log marginal likelihood function, that is: simultaneously learn the parameters θ of the deep neural network and the hyperparameters ψ of the kernel function. The training objective function is expressed as

$$\log p(y|\mathbf{X};\theta,\psi) = -\frac{1}{2}y^T K^{-1} y - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi \tag{6}$$

The deep kernel learning method presented in this paper offers several notable advantages: it extracts features from black-box functions using deep neural networks and integrates them into Gaussian process kernel functions, thereby unifying the training of deep learning models with the optimization of kernel functions within a single framework. This approach effectively combines the nonlinear feature extraction capabilities of deep neural networks with the Bayesian uncertainty inference provided by Gaussian processes. Furthermore, it facilitates the flexible integration of various types of deep neural networks and kernel functions, enabling the extension of Bayesian optimization to more complex tasks involving black-box function optimization.

3.2. Adaptive gradient-based trust region size adjustment

In the TuRBO algorithm, a heuristic strategy based on success/failure counts is adopted for the adjustment of the trust region. The idea is as follows.

$$L_{n+1} = \begin{cases} \min(2L_{n+1}, L_{\max}) & \text{if } S \geq \tau_{\text{succ}} \\ L_n/2 & \text{if } F \geq \tau_{\text{fail}} \\ L_n & \text{otherwise} \end{cases} \tag{7}$$

Where S and F respectively represent the current consecutive successful and failed assessment counts. L_n is the trust region radius. This strategy presents two significant drawbacks. Firstly, it underutilizes available information by relying solely on the count of successes and failures, thereby completely overlooking the local geometric features of the objective function. Secondly, the adjustment mechanism for the trust region radius is overly simplistic; it either expands by a fixed factor of 2 or contracts by a factor of 0.5 with each iteration. This rigid approach fails to adaptively align with the local characteristics of the current solution. To address these limitations, this paper introduces a gradient-aware dynamic adjustment strategy for modifying the trust region radius. This method employs the central difference technique to compute sparse gradients and establishes a gradient history window of size to inform the dynamic resizing of the trust region. The proposed steps for adjusting the trust region radius are detailed in Table 3.

Table 3: Steps for adjusting the trust region radius in adaptive gradient.

```
Input: Observation point set
$D_{n_0} = \{(\mathbf{x}_1, f(\mathbf{x}_1)), \dots, (\mathbf{x}_{n_0}, f(\mathbf{x}_{n_0}))\}$, candidate point set $D' = \left\{(\mathbf{x}'_j, f(\mathbf{x}'_j))\right\}_{j=1}^{q}$.
Output: The updated trust region radius $L_{n+1}$.
1: if $\min(f(\mathbf{x}')) < \min(f(\mathbf{x}))$
    $S \leftarrow S + 1, F \leftarrow 0.$
else
    $F \leftarrow F + 1, S \leftarrow 0.$
end if
2: For each sample in $D'$, calculate the change gradient of each dimension, and use the central
difference formula (Shi et al., 2023) to calculate the approximate gradient at each sample.
for $i = 1$ to $d$ do
    $$\frac{\partial f}{\partial \mathbf{x}_i} \approx \frac{f(\mathbf{x}_j + \epsilon \mathbf{e}_i) - f(\mathbf{x}_j - \epsilon \mathbf{e}_i)}{2\epsilon} \tag{8}$$
    $$G[i] \leftarrow \frac{\partial f}{\partial \mathbf{x}_i} \tag{9}$$
end for
3: Calculate the average gradient $\Delta_g$ on each sample using the following formula, where $\eta = 5$ is
used to control the sensitivity.
    $$\Delta_g \leftarrow \left\| \frac{1}{w} \sum_{i=q-w+1}^{q} g_i - \frac{1}{w} \sum_{i=q-2w+1}^{q-w} g_i \right\| \tag{10}$$
    $$\alpha \leftarrow 1 + 0.5 \cdot \tanh(\eta \cdot \Delta_g) \tag{11}$$
4: The trust region radius is adjusted by the average gradient, and the calculation formula is as follows.
    $$L_{n+1} = \begin{cases} \min(L_n \cdot (2(1+\alpha)), L_{\max}) & \text{if } S \geq \tau_{\text{succ}} \\ L_n / (2(1+\alpha)) & \text{if } F \geq \tau_{\text{fail}} \\ L_n & \text{otherwise} \end{cases} \tag{12}$$
```

In the algorithm steps in Table 3, the functions $\tanh(\cdot)$ map Δ_g to the [0,1) to achieve a smooth transition and avoid oscillations caused by stepwise abrupt changes.

3.3. Hybrid sampling strategy

Thompson posterior mean sampling (TS) is a sophisticated sampling strategy grounded in probabilistic surrogate models (Mao et al., 2025). The fundamental principle of this approach is to direct the sampling process through the mean of the posterior distribution, rather than by directly sampling random functions. Specifically, a random function $f \sim \mathcal{GP}(\mu, K)$ is drawn from the posterior distribution of the Gaussian process, and the parameter $\mathbf{x}$ that corresponds to its maximum value is chosen as the next evaluation point.

The Gaussian process upper confidence bound (GP-UCB) (Kim et al., 2024) sampling balances exploration and exploitation through the upper bound of the confidence interval strategy. The formula of GP-UCB is as follows.

$$\alpha_{\texttt{ucb}}(\mathbf{x}; D_{1:n}) = \mu_n(\mathbf{x}) + \beta\sigma_n(\mathbf{x}) \tag{13}$$

Where $\mu_n(\mathbf{x})$ and $\sigma_n(\mathbf{x})$ represent the mean and variance of the posterior distribution respectively. The hyperparameter $\beta = 0.1$ is used to balance exploration and exploitation. When β is large, the algorithm tends to explore; conversely, when β is small, the algorithm tends to exploit.

A single sampling strategy frequently results in Bayesian optimization becoming trapped in local optima throughout the iterative process. To mitigate this challenge, this paper introduces a hybrid sampling strategy that probabilistically selects between two approaches: the Gaussian process upper confidence bound (GP-UCB) strategy is employed with a probability of 0.66 to generate new evaluation points, while the Thompson sampling (TS) strategy is utilized for the remaining new evaluation points with a probability of 0.34. By combining these two distinct sampling strategies probabilistically to produce new evaluation points, we enhance their diversity, thereby facilitating the BO algorithm's ability to escape from local optima during its iterative progression.

3.4. The DKG-TBO algorithm framework

Based on the above key technologies, the steps of the DKG-TBO algorithm proposed in this paper are shown in Table 4.

Table 4: Steps of the DKG-TBO algorithm.

```
Input: The objective function $f$, the initial sampling points $n_0$ , the total number of evaluations
of the objective function $N$, and the initial trust region $TR$.
Output: The optimal solution of the objective function $\mathbf{x}^*$.
1: Determine the deep kernel learning model DKL and the GPR surrogate model, and determine
acquisition function of the Hybrid sampling strategy $TU(\mathbf{x}; D_n)$.
2: Evaluate $n_0$ samples, that is, $D_{n_0} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{n_0}, y_{n_0})\}$, $y_i = f(\mathbf{x}_i)$, $n \leftarrow n_0$
3: while $n \leq N$ do
4: The DKL training is conducted on the observation set $D_n$ to obtain the learned training kernel
parameters $\psi$, and then the GPR model is constructed.
5: $\mathbf{x}^* = \arg\min_{\mathbf{x} \in D} y(\mathbf{x})$, build $TR$ centered on $\mathbf{x}^*$.
6: Generate the next evaluation point $\mathbf{x}_{n+1}$ , and evaluate it.
$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x} \in \mathcal{X}} TU(\mathbf{x}; D_n)$, $y_{n+1} \leftarrow f(\mathbf{x}_{n+1})$.
7: Update , according to the strategy in Table 3.
8: $D_{n+1} \leftarrow D_n \cup \{(\mathbf{x}_{n+1}, y_{n+1})\}$
9: end while
10: Return the optimal solution $\mathbf{x}^*$.
```

4. Experiments

4.1. Experimental cases

This paper selects three widely utilized classical synthetic functions in the realm of Bayesian optimization: Ackley (Demo et al., 2021), Levy (Laguna and Marti, 2005), and Griewank (Toktas et al., 2024). Additionally, it examines two real-world problems: the 14D robot pushing problem (Wang et al., 2017) and the 60D rover trajectory planning with obstacle avoidance. For the synthetic functions, we set the function dimension to 60. In each iteration, a batch of sampling points is evaluated, with a batch size of 10; furthermore, the initial number of sample points for the algorithm is established at 20. The optimization algorithms compared in this study include TuRBO, TRLBO (Li et al., 2023), BO-UCB (Lalitha and Goldsmith, 2021), TPE (Chen and Seo, 2023), SAASBO (Eriksson and Jankowiak, 2021), and PSO (Jain et al., 2022). The TPE algorithm represents a Bayesian optimization approach that employs a random forest as its surrogate model. Conversely, SAASBO is an enhanced Bayesian optimization method tailored for high-dimensional problems that mitigates invalid dimensions within the parameter space by sparsifying the kernel function. The PSO algorithm stands as a classic differential evolution optimization technique. Moreover, in this experiment, uniform sampling is employed to generate batches of solution sets within the solution space, maintaining a batch size of 10. Each test function undergoes independent repetition for a total of 30 trials.

4.2. Results and Analysis

Figure 1 illustrates the average convergence curves of various algorithms applied to the 60D Ackley function. In this figure, DKG-TBO demonstrates a rapid decrease in the objective function value during the early stages of iteration, achieving an outcome that is already very close to the theoretical optimal value of 0 for the Ackley function after just 1000 evaluations. In contrast, traditional BO-UCB becomes trapped in a local optimum following these evaluations and fails to escape it in subsequent iterations. Although TuRBO and TRLBO also employ trust region strategies, DKG-TBO exhibits faster convergence and attains a lower objective function value (indicating a superior solution). SAASBO approaches the theoretical optimal value of the Ackley function later in its iterations; however, it requires nearly five times more evaluations than DKG-TBO to do so. Meanwhile, PSO shows a steady decline in its objective function value throughout the iterative process; nevertheless, the optimal solution it arrives at is significantly inferior compared to that achieved by DKG-TBO.

Figure 2 illustrates the convergence comparison of various algorithms applied to the 60D Griewank function. It is evident from Figure 2 that, with the exception of BO-UCB, all other algorithms are able to approach the theoretical optimal value of 0 for the Griewank function after 6000 evaluations. DKG-TBO, TPE, TRLBO, and TuRBO demonstrate a rapid decrease in objective function values and converge towards the theoretical optimal value (zero) during the early iterations; however, DKG-TBO requires the fewest function evaluations among them.

Figure 3 illustrates the convergence comparison of various algorithms applied to the 60D Levy function. In this figure, with the exception of PSO and BO-UCB, all other algorithms are able to approach the theoretical optimal value (zero) for the Levy function after 6000 evaluations. The algorithms that demonstrate the fastest decrease in objective function values include DKG-TBO, TPE, TRLBO, TuRBO, and SAASBO; among these, DKG-TBO requires the fewest function evaluations.

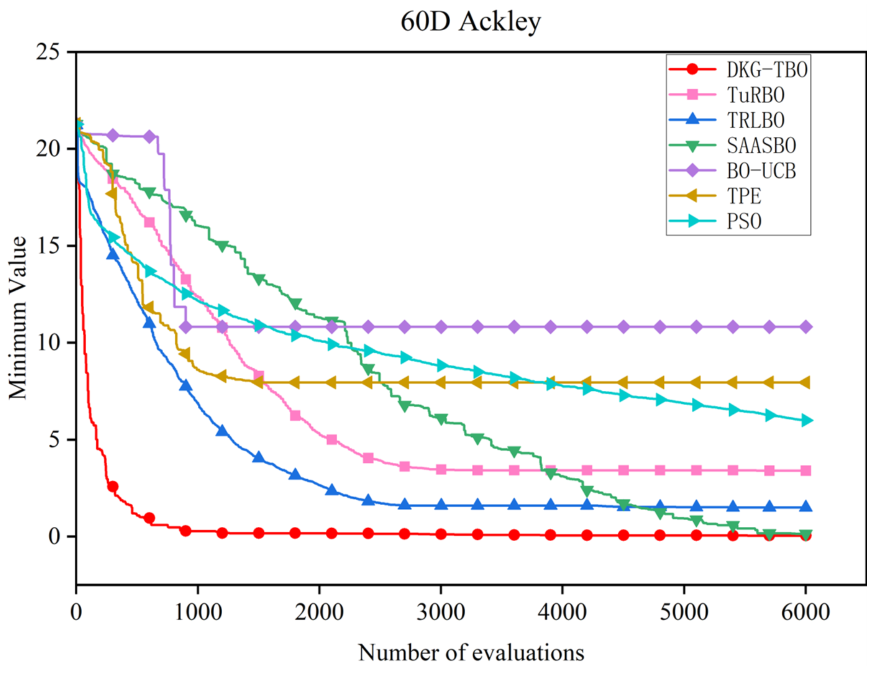

Figure 1: Comparison of the convergence of various algorithms on the 60D Ackley function.

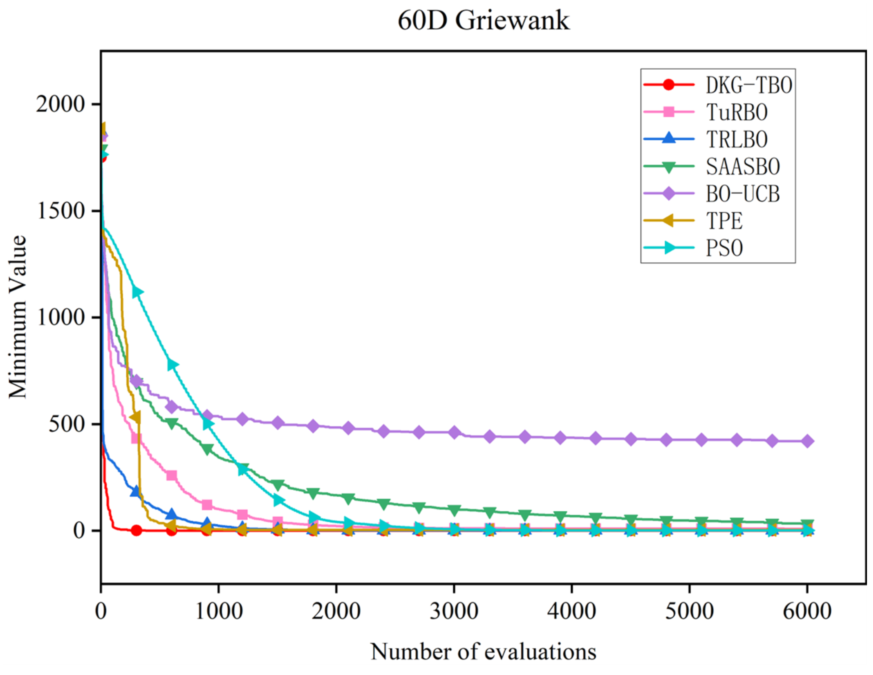

Figure 2: Comparison of the convergence of each algorithm on the 60D Griewank function.

Figure 4 presents a comparison of the convergence behaviors of various algorithms applied to the 14D robot pushing problem. This problem primarily simulates the control of a robot tasked with

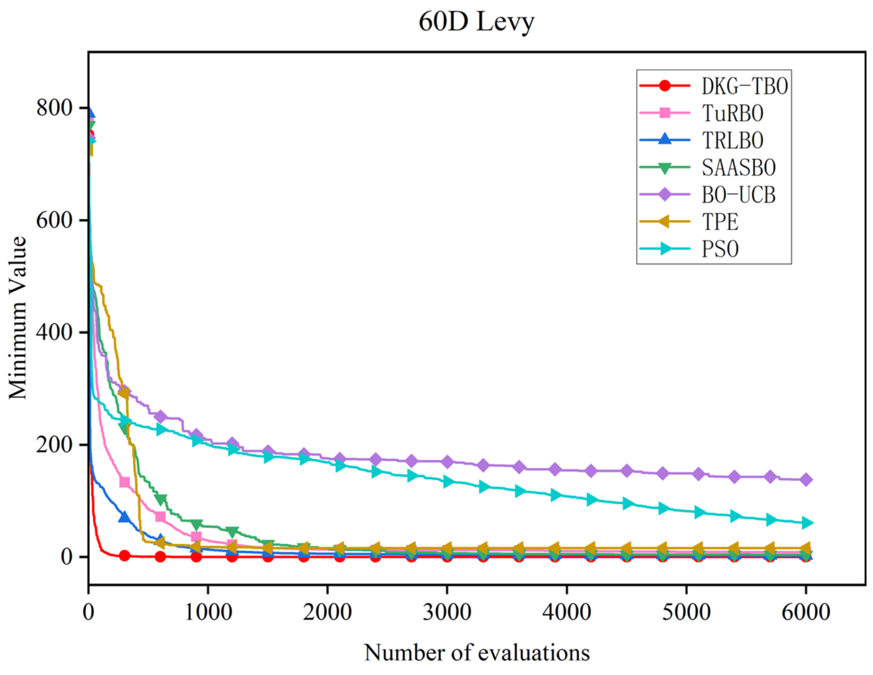

Figure 3: Comparison of the convergence of each algorithm on the 60D Levy function.

pushing two objects to a designated position, where the trajectory of the robot's movement is governed by 14 parameters (Toktas et al., 2024). The data illustrated in Figure 4 demonstrates that the DKG-TBO algorithm exhibits a steady increase in objective function value during early iterations and converges to a relatively optimal solution after approximately 2000 to 3000 evaluations. Among the algorithms compared, only TuRBO and SAASBO achieve an optimal solution that approaches that of DKG-TBO; however, they necessitate nearly four times as many evaluations as required by DKG-TBO.

Figure 5 illustrates a comparison of the convergence rates of various algorithms applied to the 60-dimensional trajectory planning problem for a Mars rover. This dynamic trajectory optimization challenge is framed within a two-dimensional map, characterized by an objective function that is non-smooth and discontinuous, as elaborated in reference (Li et al., 2022). As depicted in Figure 5, after conducting 20,000 function evaluations, the optimal values achieved by DKG-TBO, SAASBO, TuRBO, and PSO are relatively comparable. However, it is noteworthy that DKG-TBO attains superior solutions with fewer evaluations than its counterparts.

A review of the convergence curves presented in Figures 1 to 5 indicates that DKG-TBO exhibits exceptional convergence across five test cases, each with distinct characteristics. The algorithm demonstrates a reduced number of evaluation iterations, rapid convergence of the objective function, and superior quality solutions. These findings suggest that the algorithm proposed in this paper possesses remarkable optimization capabilities. Specifically, the GPR model integrated within the proposed algorithm showcases effective feature recognition and fitting abilities for various black-box functions, while the acquisition function introduced offers high search efficiency within the solution space. Consequently, this algorithm can swiftly approach the optimal solution to problems with minimal evaluation iterations.

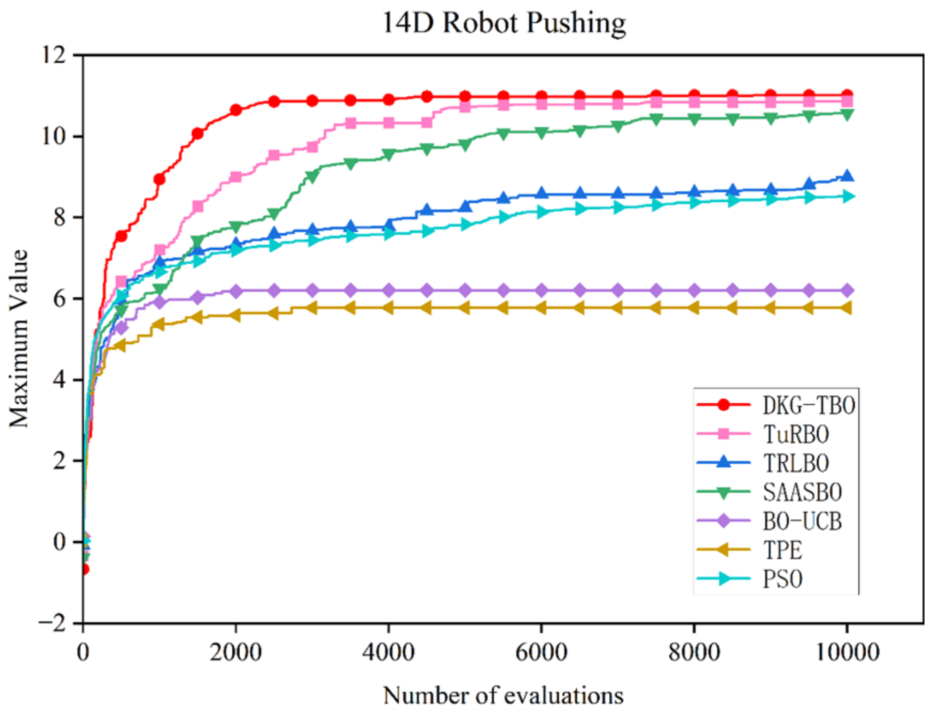

Figure 4: Comparison of the convergence of various algorithms on the 14D robot pushing problem.

Table 5 presents the computational results of various algorithms applied to three 60-dimensional test functions. It summarizes the average values and standard deviations derived from 30 independent repeated experiments for each algorithm, with a maximum evaluation count of 6000 per experiment. The values highlighted in bold in Table 5 indicate the best results achieved. From Table 5, we can draw the following observations: (1) For the Ackley test function, DKG-TBO outperforms all other algorithms, achieving superior results. The average performance metrics of DKG-TBO, TRLBO, and BO-UCB are relatively close; however, despite all three employing trust region techniques, DKG-TBO demonstrates mean accuracy that is 97.03% and 99.06% higher than those of TuRBO and TRLBO respectively. (2) In relation to the Griewank test function, while PSO achieves the highest mean value overall, it is noteworthy that the difference between the mean values of DKG-TBO and PSO is merely 8.18%, indicating their proximity in performance. Again here too—despite utilizing trust region strategies—DKG-TBO exhibits a mean accuracy that surpasses TuRBO and TRLBO by margins of 88.6% and 97.1%, respectively. (3) Concerning the Levy test function, DKG-TBO records the highest mean value among all tested algorithms as well; its mean accuracy exceeds that of TuRBO and TRLBO by significant margins of 57.09% and 78.65%, respectively. Based on these findings presented in Table 5, it can be concluded that DKG-TBO demonstrates superior computational accuracy across all evaluated algorithms on these three distinct 60D test functions—highlighting its robust optimization capabilities.

Table 6 presents the computational results of various algorithms applied to the 14D robot pushing problem. The data in Table 6 represent the average values and standard deviations derived from 30 independent and repeated experiments conducted by each algorithm, with each experiment in-

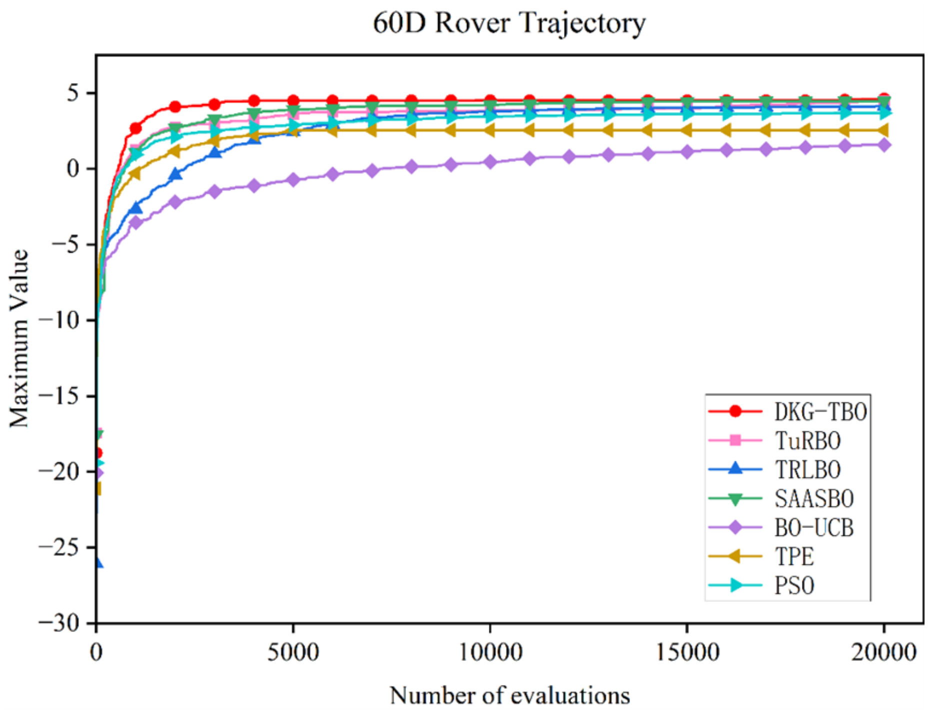

Figure 5: Comparison of the convergence of various algorithms on the 60D rover trajectory problem.

Table 5: Computational results of each algorithm on three 60D test functions.

Test function	Indicator	DKG-TBO	TuRBO	TRLBO	SAASBO	BO-UCB	TPE	PSO
Ackley	Mean	$1.01e-01$	$3.40e+00$	$1.08e+01$	$1.40e-01$	$1.08e+01$	$7.94e+00$	$6.00e+00$
	Deviation	$1.18e-01$	$2.34e-01$	$7.79e-01$	$6.42e-01$	$7.79e-01$	$1.79e+00$	$3.69e-01$
Griewank	Mean	$1.10e+00$	$8.21e+00$	$3.79e+01$	$3.30e+01$	$4.20e+02$	$3.21e+00$	$1.01e+00$
	Deviation	$8.72e+00$	$8.30e+00$	$1.12e+01$	$2.05e+01$	$4.92e+01$	$4.36e-01$	$3.24e-02$
Levy	Mean	$3.63e+00$	$8.46e+00$	$1.70e+01$	$3.85e+00$	$1.38e+02$	$1.57e+01$	$6.06e+01$
	Deviation	$2.97e+00$	$3.13e+00$	$3.88e+00$	$3.94e+00$	$1.55e+01$	$5.34e+00$	$1.59e+01$

volving 10,000 evaluations. In this table, the means of DKG-TBO, TuRBO, TRLBO, and SAASBO are relatively close; however, DKG-TBO exhibits the highest mean performance. Specifically, the mean value for DKG-TBO is 0.9% greater than that of TuRBO, indicating comparable computational accuracy between these two algorithms in this instance. Conversely, DKG-TBO's mean is significantly higher—by 18.09%—than that of TRLBO. The findings presented in Table 6 suggest that DKG-TBO demonstrates strong performance on the 14D robot pushing problem as well.

Table 6: Computational results of each algorithm on the 14D robot pushing problem.

Indicator	DKG-TBO	TuRBO	TRLBO	SAASBO	BO-UCB	TPE	PSO
Mean	**1.10e+01**	1.09e+01	9.01e+00	1.05e+01	6.20e+00	5.78e+00	8.52e+00
Deviation	**3.35e-01**	4.23e-01	6.05e-01	6.40e-01	3.71e+00	6.01e-01	1.80e+00

Table 7 presents the computational results of various algorithms applied to the 60D rover trajectory planning problem. The data in Table 7 represent the average values and standard deviations derived from 30 independent and repeated experiments conducted by each algorithm, with each experiment evaluated a total of 20,000 times. From Table 7, it is evident that DKG-TBO achieves the highest mean value among all algorithms, accompanied by the smallest standard deviation of 2.35e-02. Notably, the mean value of DKG-TBO is approximately 3.15% greater than that of TuRBO, which ranks second overall.

Table 7: Computational results of various algorithms on the 60D rover trajectory planning problem.

Indicator	DKG-TBO	TuRBO	TRLBO	SAASBO	BO-UCB	TPE	PSO
Mean	**4.59e+00**	4.45e+00	4.12e+00	4.47e+00	1.57e+00	2.55e-02	3.69e+00
Deviation	**2.35e-02**	3.34e-01	7.70e-01	9.02e-01	1.02e+00	5.17e-01	8.19e-01

Based on the findings presented in Tables 5 through 7, it is evident that DKG-TBO demonstrates commendable computational accuracy across five high-dimensional nonlinear black-box optimization problems. This suggests that the surrogate model employed by DKG-TBO effectively captures the characteristics inherent to black-box challenges, while its acquisition function adeptly navigates the solution space to yield high-quality solutions. Consequently, it can be concluded that the enhancements proposed for the BO algorithm in this paper are both feasible and effective.

5. Conclusions

This paper addresses the performance bottleneck of traditional Bayesian optimization algorithms in high-dimensional black-box optimization problems and proposes a novel trust region Bayesian optimization algorithm (DKG-TBO). In the design of the surrogate model for Bayesian optimization, we propose to integrate the feature extraction capabilities of deep neural networks with the nonlinear mapping advantages offered by kernel methods. This integration enhances the Gaussian process regression model, thereby improving its feature extraction and modeling capabilities for high-dimensional black-box functions. Simultaneously, an adaptive gradient trust region adjustment strategy along with a TS-UCB sampling strategy is introduced to effectively balance the trade-off between exploration and exploitation during the acquisition function's search within the solution space. This enhancement improves the algorithm's ability to escape local optima while bolstering its global search optimization capability. Through evaluation on five high-dimensional test cases exhibiting diverse characteristics, DKG-TBO demonstrates exceptional performance in terms of iterative convergence and computational accuracy. These results validate that the improved BO algorithm proposed in this study is both feasible and effective.

Although this study has achieved the aforementioned successes, several deficiencies remain. (1) While there are numerous representative models of deep neural networks, this paper does not provide an in-depth comparison of their effectiveness concerning the Bayesian optimization algorithm and their respective characteristics. (2) The sampling strategy presented in this article still has potential for enhancement. (3) The efficacy of the proposed algorithm in other more complex application scenarios requires further validation. These issues remain to be further discussed in future research.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 52275003, in part by the National Key Research and Development Program of China under Grant 2023YFB4704000; Tianshan Talent Training Program (NO. 2023TSYCLJ0052)

References

H. T. Bunn, J. V. Gobburu, and L. M. Floryance. Bayesian model-guided antimicrobial therapy in pediatrics. *Frontiers in Pharmacology*, 14:1118771, 2023. doi: 10.3389/fphar.2023.1118771.

C. Chen and H. Seo. Prediction of rock mass class ahead of tbm excavation face by ml and dl algorithms with bayesian tpe optimization and shap feature analysis. *Acta Geotechnica*, 18(7): 3825–3848, 2023. doi: 10.1007/s11440-023-01943-0.

N. Demo, M. Tezzele, and G. Rozza. A supervised learning approach involving active subspaces for an efficient genetic algorithm in high-dimensional optimization problems. *SIAM Journal on Scientific Computing*, 43(3):B831–B853, 2021. doi: 10.1137/20M1340618.

B. Eric, N. Freitas, and A. Ghosh. Active preference learning with discrete choice data. In *Advances in Neural Information Processing Systems*, volume 20, 2007.

D. Eriksson and M. Jankowiak. High-dimensional bayesian optimization with sparse axis-aligned subspaces. In *Uncertainty in Artificial Intelligence*, pages 493–503. PMLR, 2021.

D. Eriksson, M. Pearce, J. Gardner, et al. Scalable global optimization via local bayesian optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

P. I. Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

F. Gisperg, R. Klausser, M. Elshazly, and et al. Bayesian optimization in bioprocess engineering—where do we stand today? *Biotechnology and Bioengineering*, 2025. doi: 10.1002/bit.28343.

S. Greenhill, S. Rana, S. Gupta, and et al. Bayesian optimization for adaptive experimental design: A review. *IEEE Access*, 8:13937–13948, 2020. doi: 10.1109/ACCESS.2020.2975868.

F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers*, pages 507–523. Springer, 2011.

M. Jain, V. Saihjpal, N. Singh, and et al. An overview of variants and advancements of pso algorithm. *Applied Sciences*, 12(17):8392, 2022. doi: 10.3390/app12178392.

H. Kim, D. Sanz-Alonso, and R. Yang. Optimization on manifolds via graph gaussian processes. *SIAM Journal on Mathematics of Data Science*, 6(1):1–25, 2024. doi: 10.1137/23M1531255.

M. Laguna and R. Marti. Experimental testing of advanced scatter search designs for global optimization of multimodal functions. *Journal of Global Optimization*, 33:235–255, 2005. doi: 10.1007/s10898-005-2981-2.

A. Lalitha and A. Goldsmith. Bayesian algorithms for decentralized stochastic bandits. *IEEE Journal on Selected Areas in Information Theory*, 2(2):564–583, 2021. doi: 10.1109/JSAIT.2021.3056773.

Q. Li, A. Fu, W. Wei, and et al. A trust region based local bayesian optimization without exhausted optimization of acquisition function. *Evolving Systems*, 14(5):839–858, 2023. doi: 10.1007/s12237-023-01083-1.

W. Li, W. Li, L. Cheng, and et al. Trajectory optimization with complex obstacle avoidance constraints via homotopy network sequential convex programming. *Aerospace*, 9(11):720, 2022. doi: 10.3390/aerospace9110720.

J. Mao, Y. Zhu, G. Chen, and et al. Automated conceptual design of mechanisms based on thompson sampling and monte carlo tree search. *Applied Soft Computing*, 170:112659, 2025. doi: 10.1016/j.asoc.2023.112659.

M. McIntire, D. Ratner, and S. Ermon. Sparse gaussian processes for bayesian optimization. In *UAI*, volume 16, pages 517–526, 2016.

H.-J. M. Shi, M. Qiming Xuan, F. Oztoprak, and et al. On the numerical performance of finite-difference-based methods for derivative-free optimization. *Optimization Methods and Software*, 38(2):289–311, 2023. doi: 10.1080/10556788.2022.2138546.

F. Toktas, U. Erkan, and Z. Yetgin. Cross-channel color image encryption through 2d hyperchaotic hybrid map of optimization test functions. *Expert Systems with Applications*, 249:123583, 2024. doi: 10.1016/j.eswa.2024.123583.

Z. Wang, C. Li, S. Jegelka, and et al. Batched high-dimensional bayesian optimization via structural kernel learning. In *International Conference on Machine Learning*, pages 3656–3664. PMLR, 2017.

Z. Wang, C. Gehring, P. Kohli, and et al. Batched large-scale bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, pages 745–754. PMLR, 2018.

A. G. Wilson, Z. Hu, R. Salakhutdinov, et al. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378. PMLR, 2016.

J. Wu, X.-Y. Chen, H. Zhang, and et al. Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology*, 17(1):26–40, 2019. doi: 10.1109/JEST.2019.2900568.

A. Yamamoto, A. Yamanaka, K. Iida, and et al. Integrating machine learning with advanced processing and characterization for polycrystalline materials: a methodology review and application to iron-based superconductors. *Science and Technology of Advanced Materials*, 26(1):2436347, 2025. doi: 10.1080/14686996.2025.2436347.

J. Yang, Z. Wu, Z. Wang, and et al. Enhanced anisotropic radius basis function metamodel based on recursive evolution latin hypercube design and fast k-fold cross-validation. *Structural and Multidisciplinary Optimization*, 66(7):169, 2023. doi: 10.1007/s00158-023-03508-8.

T. Yu and H. Zhu. Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689*, 2020.

Y.-x. Yuan. A review of trust region algorithms for optimization. In *ICIAM*, volume 99, pages 271–282, 2000.