

Meta-Black-Box Optimization with Bi-Space Landscape Analysis and Dual-Control Mechanism for SAEA

**Yukun Du¹, Haiyue Yu^{1*}, Xiaotong Xie¹, Yan Zheng¹,
Lixin Zhan¹, Yudong Du², Chongshuang Hu¹, Boxuan Wang¹, Jiang Jiang¹**

¹National University of Defense Technology,

²Xinxiang University,

duyukun-nudt@outlook.com, yuhaiyue09@nudt.edu.cn, xiexiaotong20@nudt.edu.cn,

zhengyan24@nudt.edu.cn, zhanlixin98@outlook.com, duyudong668@outlook.com,

huchongshuang@foxmail.com, boxuanwang24@nudt.edu.cn, jiangjiangnudt@nudt.edu.cn

Abstract

Surrogate-Assisted Evolutionary Algorithms (SAEAs) are widely used for expensive Black-Box Optimization. However, their reliance on rigid, manually designed components such as infill criteria and evolutionary strategies during the search process limits their flexibility across tasks. To address these limitations, we propose Dual-Control Bi-Space Surrogate-Assisted Evolutionary Algorithm (DB-SAEA), a Meta-Black-Box Optimization (MetaBBO) framework tailored for multi-objective problems. DB-SAEA learns a meta-policy that jointly regulates candidate generation and infill criterion selection, enabling dual control. The bi-space Exploratory Landscape Analysis (ELA) module in DB-SAEA adopts an attention-based architecture to capture optimization states from both true and surrogate evaluation spaces, while ensuring scalability across problem dimensions, population sizes, and objectives. Additionally, we integrate TabPFN as the surrogate model for accurate and efficient prediction with uncertainty estimation. The framework is trained via reinforcement learning, leveraging parallel sampling and centralized training to enhance efficiency and transferability across tasks. Experimental results demonstrate that DB-SAEA not only outperforms state-of-the-art baselines across diverse benchmarks, but also exhibits strong zero-shot transfer to unseen tasks with higher-dimensional settings. This work introduces the first MetaBBO framework with dual-level control over SAEAs and a bi-space ELA that captures surrogate model information.

Introduction

Surrogate-Assisted Evolutionary Algorithms (SAEAs) are effective approaches for solving expensive Black-Box Optimization (BBO) problems (Cai, Gao, and Li 2019; Jin 2011). By introducing surrogate models, such as Gaussian Processes (GPs), to approximate the true objective function and combining them with evolutionary algorithms for global search (Lim et al. 2009), SAEAs can deliver high-quality solutions while reducing the number of expensive evaluations (He et al. 2023; Zhan and Xing 2021). However, existing SAEAs still rely on rigid, manually crafted

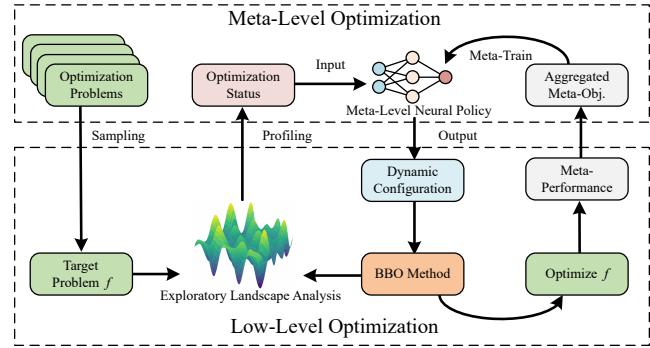


Figure 1: The general workflow of MetaBBO

components, such as infill criterion design and evolutionary strategy configuration, which limits their flexibility (Akbari and Kazerooni 2020; Liu et al. 2021; Zhang, Chen, and Cheng 2020). As a result, SAEAs struggle to dynamically adjust their components based on task characteristics and the evolving requirements of different search phases, thereby limiting their transferability and adaptability.

Beyond the fixed design of conventional SAEAs, Meta-Black-Box Optimization (MetaBBO) provides a more general, learning-based framework for adaptive optimization across diverse tasks and dynamic search conditions. By formulating algorithm configuration as a data-driven control process (Wu and Wang 2022; Lange et al. 2023), MetaBBO reduces manual intervention and improves transferability and adaptability across tasks and search phases. As shown in Figure 1, it follows a bi-level structure (Ma et al. 2023), where the meta-level gathers feedback from the low-level optimizer (i.e., BBO Method in Figure 1) via Exploratory Landscape Analysis (ELA) (Mersmann et al. 2011; Renau et al. 2019, 2020). This feedback is then utilized by a meta-policy to dynamically adjust key components, such as optimizer selection and search strategies, thereby forming a closed-loop adaptation mechanism (Tsaban et al. 2022; Akbari et al. 2019; Guo et al. 2025; Ning et al. 2018).

Despite the increasing attention to MetaBBO, most existing approaches still rely on traditional evolutionary algorithms as low-level optimizers (Ma et al. 2025b; Yang

*Corresponding author.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

et al. 2025). For instance, R2-RLMOEA (Tahernezhad-Javazm et al. 2024) employs a reinforcement learning agent to switch among several evolutionary algorithms, while LDE (Sun et al. 2021) leverages an LSTM network to adjust key parameters—such as the mutation factor and crossover rate—under varying search conditions. Although these works highlight the potential of MetaBBO for adaptive control, the integration of SAEAs into the MetaBBO paradigm remains largely unexplored. This gap is primarily due to several fundamental challenges:

1) **Difficulty in landscape-aware representation.** In MetaBBO, ELA characterizes current search behavior via landscape-aware state representations that guide meta-policy decisions. However, constructing informative state representations remains difficult, especially under surrogate-assisted settings. The state should reflect both the population evaluated by the true objective and surrogate-derived signals such as predictive uncertainty, which are often high-dimensional, correlated, and hard to encode for effective learning and control.

2) **Lack of unified meta-level control over key components of SAEA.** Existing MetaBBO methods rarely investigate how to jointly control the key components of SAEA, including the surrogate model, evolutionary strategy, and infill criterion. For example, DRL-SAEA (Shao, Tian, and Zhang 2025), integrates SAEA as the low-level optimizer but restrict control to surrogate model management, without extending control to other essential modules. Designing a unified meta-policy that manages multiple tightly coupled modules and adapts to changing search conditions remains a challenging problem.

3) **Limitations of surrogate models.** Traditional surrogate models such as Gaussian Processes (GPs) suffer from high computational cost and poor scalability with dimensionality. In high-dimensional problems, they often produce unreliable predictions, limiting their ability to support meta-policy learning (Seeger 2004; Noack et al. 2021). Moreover, the high training cost of GPs makes them impractical for MetaBBO frameworks, which typically require frequent model updates across many tasks.

To promote the integration of SAEAs into the MetaBBO paradigm, we propose DB-SAEA, a novel MetaBBO framework for expensive multi-objective problems, which employs a SAEA as the low-level optimizer. DB-SAEA is designed to address the challenges by integrating the following techniques:

1) **Bi-space neural network-based ELA.** We design a bi-space ELA module using multiple Two-Stage Attention (Ts-Attn) blocks to extract landscape-aware features from both true and surrogate evaluation spaces. This design integrates surrogate model signals to construct optimization states. The resulting representations enable scalable landscape modeling across varying problem dimensions, population sizes, and numbers of objectives, and are jointly trained with the meta-policy to couple feature extraction with decision control.

2) **Dual-control mechanism for SAEA.** We propose a dual-control mechanism where the meta-policy jointly governs infill criterion selection and controls the evolutionary

algorithm by deciding whether candidate solutions should undergo true evaluation. In each iteration, the evolutionary algorithm generates candidate solutions, and the meta-policy decides whether to proceed with true evaluation or continue surrogate-assisted search. If true evaluation is chosen, the meta-policy further selects the most suitable infill criterion based on the current optimization state.

3) **Efficient surrogate modeling with TabPFN.** To address the inefficiency and poor scalability with dimensionality of GPs, we adopt TabPFN (Hollmann et al. 2025), a transformer-based probabilistic estimator tailored for tabular data. TabPFN supports fast inference without task-specific training, while providing accurate objective prediction and uncertainty estimation.

The main contributions of this work are as follows. 1) we are first to propose a dynamic, multi-objective landscape-aware representation method that is capable of capturing surrogate model information. 2) We overcome the limitations of single-component control by developing a dual-control mechanism, establishing a novel control paradigm for MetaBBO. 3) We integrate TabPFN into the SAEA framework as an efficient surrogate model for fast and accurate prediction with uncertainty estimation.

Backgrounds

Neural Network-based ELA

Recent advances in neural network-based ELA have introduced methods such as Deep-ELA (Seiler, Kerschke, and Trautmann 2025) and NeurELA (Ma et al. 2025a), both leveraging attention mechanisms to model optimization landscapes. Deep-ELA, pre-trained on large-scale static problems, generates one-shot global representations but lacks adaptability to evolving optimization states and models only interactions across sample points. NeurELA mitigates these limitations by introducing a two-stage attention mechanism that captures both inter-solution and inter-dimension interactions, enhancing its representational capacity in dynamic MetaBBO tasks. However, it remains restricted to single-objective settings and, like Deep-ELA, does not incorporate feedback from surrogate models, limiting its ability to characterize the surrogate evaluation space.

To address the aforementioned limitations, we propose a bi-space neural network-based ELA tailored for multi-objective optimization, serving as a core component of the DB-SAEA framework. Building on existing neural ELA methods, our approach extends them by incorporating both true and surrogate evaluation spaces, enabling dynamic, landscape-aware representations across multiple objectives. Here, we construct the landscape representations of both the true and surrogate evaluation spaces using their respective populations. In an m -objective optimization problem, these populations are defined independently for each space:

$$\begin{aligned} \mathcal{P}_{\text{true}} &= \left\{ (\mathbf{x}_i, \mathbf{y}_i) \mid \mathbf{y}_i = \mathbf{f}(\mathbf{x}_i), i = 1, \dots, n_{\text{true}} \right\}, \\ \mathcal{P}_{\text{sur}} &= \left\{ (\mathbf{x}_i, \hat{\mathbf{y}}_i, \hat{\sigma}_i) \mid \hat{\mathbf{y}}_i = \hat{\mathbf{f}}(\mathbf{x}_i), i = 1, \dots, n_{\text{sur}} \right\}, \end{aligned} \quad (1)$$

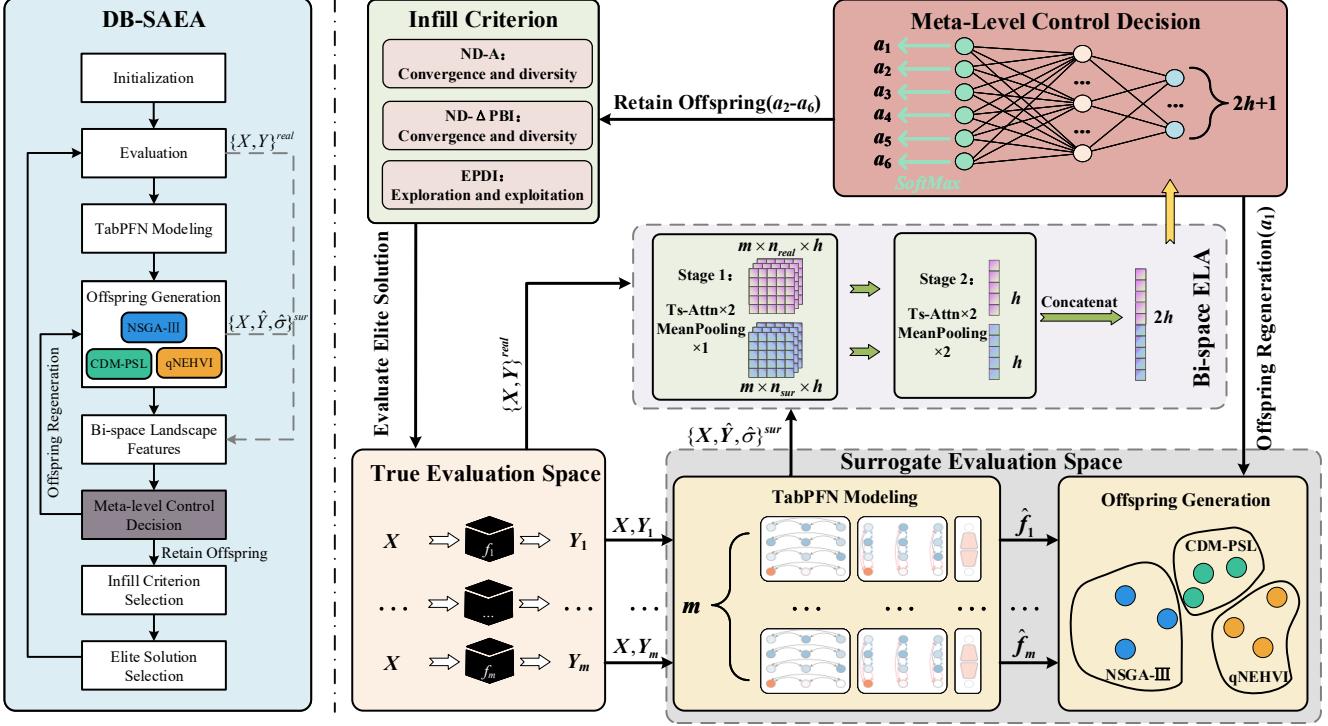


Figure 2: The framework of DB-SAEA. (right) DB-SAEA generates candidate solutions using hybrid sampling (NSGA-III, CDM-PSL, qNEHVI), evaluates them with TabPFNs, and extracts landscape-aware features via bi-space ELA. The meta-policy determines whether to perform true evaluation or continue surrogate-assisted search. Upon true evaluation, elite solutions are selected and $\mathcal{P}_{\text{true}}$ is updated.

where $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$, and $\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i) \in \mathbb{R}^m$. In the surrogate evaluation space, $\hat{\mathbf{y}}_i = \hat{\mathbf{f}}(\mathbf{x}_i) \in \mathbb{R}^m$ denotes the surrogate model’s prediction for the i -th solution, and $\hat{\sigma}_i \in \mathbb{R}_+^m$ indicates the corresponding predictive uncertainty. Notably, the true evaluation space contains all solutions evaluated by the true objectives, while the surrogate evaluation space consists of candidate solutions generated at the current step and evaluated by surrogate models.

TabPFN

This work is the first to adopt TabPFN (Tabular Prior-Data Fitted Network) as a surrogate model for optimization. TabPFN is a transformer-based probabilistic estimator that enables high-accuracy, low-latency one-shot inference without requiring task-specific training (Hollmann et al. 2025, 2023; Müller et al. 2022), making it highly suitable for expensive black-box optimization. TabPFN outputs a discrete probability distribution over predefined value intervals. Specifically, for a candidate solution $\mathbf{x}_i \in \mathbb{R}^d$, the model returns a probability vector:

$$\mathbf{p}_i = \text{TabPFN}(\mathbf{x}_i) = [p_1^{(i)}, \dots, p_K^{(i)}] \in [0, 1]^K, \quad (2)$$

where $\sum_{k=1}^K p_k^{(i)} = 1$, K denotes the number of predefined value intervals, and $p_k^{(i)}$ represents the predicted probability that the objective value of the i -th solution falls into the k -th interval.

Methodology

Overview

We propose a novel MetaBBO framework, DB-SAEA (see Algorithm 1 and Figure 2), where a SAEA serves as the low-level optimizer. Initially, solutions generated via Latin Hypercube Sampling (LHS) are evaluated using true objective functions $\mathbf{f}(\cdot)$, and these are used to construct multiple TabPFN surrogate models. Candidate solutions are then generated using a hybrid sampling strategy that integrates NSGA-III (Deb and Jain 2013), CDM-PSL (Li et al. 2025), and qNEHVI (Daulton, Balandat, and Bakshy 2020), and are evaluated by the surrogate models. A bi-space ELA Λ_{θ_Λ} constructs landscape-aware representations from both true and surrogate evaluation spaces. These representations are provided to the meta-policy π_{θ_π} , which determines whether to proceed with true evaluations or discard the candidates and resample. If the candidates are selected, the meta-policy chooses an infill criterion and identifies elite solutions for true evaluation by $\mathbf{f}(\cdot)$, followed by surrogate model updates. This process repeats until the evaluation budget is exhausted.

Bi-space Neural Network-based ELA

The ELA module of DB-SAEA first processes decision variables and objective values through a Population Information Embedding (PIE) module, which includes normaliza-

Algorithm 1: DB-SAEA

Input: Multi-objective black-box function $f(\cdot)$, evaluation budget FE_{\max} , ELA module Λ_{θ_Λ} , mata-policy π_{θ_π}
Output: Evaluated solutions $\mathcal{P}_{\text{true}}$

- 1: $X \leftarrow$ Generate an initial set of N solutions using LHS
- 2: $Y \leftarrow f(X)$
- 3: Set $\mathcal{P}_{\text{true}} \leftarrow (X, Y)$
- 4: $\hat{f}(\cdot) \leftarrow f(\cdot)$ is approximated by TabPFNs
- 5: Set evaluation counter $t \leftarrow N$
- 6: **while** $t < FE_{\max}$ **do**
- 7: Generate offspring population X_{off}
- 8: $\mathcal{P}_{\text{sur}} \leftarrow (X_{\text{off}}, \hat{f}(X_{\text{off}}), \hat{\sigma}_{\text{off}})$
- 9: Landscape-aware state $\mathbf{z}_t \leftarrow \Lambda_{\theta_\Lambda}(\mathcal{P}_{\text{true}}, \mathcal{P}_{\text{sur}})$
- 10: Optimization state $\mathbf{s}_t \leftarrow [\mathbf{z}_t, t/FE_{\max}]$
- 11: Obtain an action $a_t \leftarrow \pi_{\theta_\pi}(\mathbf{s}_t)$
- 12: **if** $a_t \neq a_1$ **then**
- 13: Select elite solutions $X^* \subset X_{\text{off}}$ based on the infill criterion specified by a_t
- 14: $Y^* \leftarrow f(X^*)$
- 15: $\mathcal{P}_{\text{true}} \leftarrow \mathcal{P}_{\text{true}} \cup (X^*, Y^*)$ and update $\hat{f}(\cdot)$
- 16: $t \leftarrow t + |X^*|$
- 17: **end if**
- 18: **end while**

tion and generates structured input representations. It then proceeds in two stages, where each stage separately applies a Ts-Attn block to both the true and surrogate evaluation spaces, enabling the extraction of informative features from each space independently.

Population Information Embedding PIE module first applies min-max normalization to both $\mathcal{P}_{\text{true}}^t$ and $\mathcal{P}_{\text{sur}}^t$ at time step t . Specifically, for the decision vectors $\{\mathbf{x}_i\}_{i=1}^{n_{\text{true}}}$ and $\{\mathbf{x}_i\}_{i=1}^{n_{\text{sur}}}$, normalization is performed against the search range. For the objective values $\{\mathbf{y}_i\}_{i=1}^{n_{\text{true}}}$, $\{\hat{\mathbf{y}}_i\}_{i=1}^{n_{\text{sur}}}$, and the surrogate uncertainties $\{\hat{\sigma}_i\}_{i=1}^{n_{\text{sur}}}$, normalization is based on the extremum values observed in $\mathcal{P}_{\text{true}}^t$ and $\mathcal{P}_{\text{sur}}^t$, respectively. This ensures unified representation and generalization by scaling all values to $[0, 1]$. For a d -dimensional optimization problem with m objectives, the normalized observations from $\mathcal{P}_{\text{true}}^t$ and $\mathcal{P}_{\text{sur}}^t$ are then reorganized:

$$\begin{aligned} M_{\text{true}}^t &= \{\{\{x_{i,j}^t, y_{i,obj}^t\}_{i=1}^{n_{\text{true}}}\}_{j=1}^d\}_{obj=1}^m, \\ M_{\text{sur}}^t &= \{\{\{x_{i,j}^t, \hat{y}_{i,obj}^t, \hat{\sigma}_{i,obj}^t\}_{i=1}^{n_{\text{sur}}}\}_{j=1}^d\}_{obj=1}^m, \end{aligned} \quad (3)$$

where $M_{\text{true}}^t \in \mathbb{R}^{m \times d \times n_{\text{true}} \times 2}$ and $M_{\text{sur}}^t \in \mathbb{R}^{m \times d \times n_{\text{sur}} \times 3}$. Then, $M_{\text{true}}^t, M_{\text{sur}}^t$ linearly projected via embedding matrices $W_{\text{emb}}^{\text{true}} \in \mathbb{R}^{2 \times h}$ and $W_{\text{emb}}^{\text{sur}} \in \mathbb{R}^{3 \times h}$ to obtain $E_{\text{true}}^t, E_{\text{sur}}^t \in \mathbb{R}^{m \times d \times n \times h}$. Here, h represents the hidden dimension.

Stage One: Multi-objective Feature Extraction In the first stage, the model extracts per-objective, individual-level landscape representations from both the true and surrogate spaces using two Ts-Attn blocks, each applied across all objectives within its respective space. The Attn block adopts the Transformer architecture (Vaswani et al. 2017) with layer normalization (Ba, Kiros, and Hinton 2016) replacing

batch normalization (Ioffe and Szegedy 2015). For the obj -objective, Ts-Attn takes $E_{\text{obj}}^t \in \mathbb{R}^{d \times n \times h}$ and enables information interaction across both individuals and dimensions. **1) Cross-individual attention:** For each evaluation space, we apply an Attn block to allow solutions within the respective population to exchange information along the same dimension. **2) Cross-dimension attention:** In each evaluation space, the Cross-individual attention yields m representations, one for each objective. We then apply a second Attn block to these representations to enable information exchange across dimensions within each solution. This involves transposing the output to shape $n \times d \times h$ and injecting positional encodings to preserve the dimensional order of each solution. **3) Pooling over problem dimensions:** For each objective, the output from the Ts-Attn module is aggregated via mean pooling along the dimension axis (i.e., over problem variables). The resulting are then stacked across all objectives to construct the multi-objective, individual-level landscape representations $S_{\text{true}}^t \in \mathbb{R}^{m \times n_{\text{true}} \times h}$ and $S_{\text{sur}}^t \in \mathbb{R}^{m \times n_{\text{sur}} \times h}$.

Stage Two: Cross-objective Feature Aggregation The representations S_{true}^t and S_{sur}^t are further refined using two additional Ts-Attn blocks, each comprising a Cross-individual and a Cross-objective attention mechanism. Unlike Stage One, we perform mean pooling along both the individual and objective axes to extract unified global representations, resulting in $s_{\text{true}}^t, s_{\text{sur}}^t \in \mathbb{R}^h$. These two vectors are then concatenated into a unified $2h$ -dimensional landscape-aware representation $\mathbf{z}_t = [s_{\text{true}}^t, s_{\text{sur}}^t] \in \mathbb{R}^{2h}$. The highly parallelizable attention-based architecture ensures that DB-SAEA remains scalable with respect to problem dimensionality, population size, and the number of objectives. Additional details are provided in **Appendix A.1**.

Low-level Optimizer Based on SAEA

Surrogate Model For each objective function f_{obj} , we set an independent TabPFN model as a surrogate. Given a solution \mathbf{x}_i , the model returns a discrete probability distribution over K bins defined by edges $\{b_k\}_{k=0}^K$. Let $\{p_k^{(i,obj)}\}_{k=1}^K$ denote the predicted probabilities corresponding to each bin for the i -th solution on the obj -th objective. To compute the scalar prediction and its uncertainty for the obj -th objective, we define the midpoint of each bin as $\mu_k = (b_{k-1} + b_k)/2$ and estimate the predicted value and standard deviation of \mathbf{x}_i using the following equations:

$$\begin{aligned} \hat{y}_{i,obj} &= \sum_{k=1}^K p_k^{(i,obj)} \cdot \mu_k, \\ \hat{\sigma}_{i,obj} &= \sqrt{\sum_{k=1}^K p_k^{(i,obj)} (\mu_k - \hat{y}_{i,obj})^2}. \end{aligned} \quad (4)$$

Evolutionary Algorithm In the evolutionary algorithm, we incorporate multiple candidate generation strategies, including NSGA-III, CDM-PSL, and qNEHVI. The effectiveness of hybrid sampling has been validated in prior studies (Li et al. 2025; Huixiang, Wenyin, and Ling 2021; Liu and

Wang 2023). The hybrid design in DB-SAEA, combining NSGA-III for global exploration with CDM-PSL and qNEHVI for focused exploitation, enhances both the diversity and structural richness of candidate solutions.

It is worth noting that only NSGA-III is used for candidate generation during training, while the hybrid strategy is applied in testing. Since CDM-PSL and qNEHVI incur high computational costs, they are unsuitable for repeated training use. Moreover, DB-SAEA focuses on controlling candidate selection based on the optimization state rather than configuring the evolutionary algorithm. Experimental results confirm that training with NSGA-III alone is sufficient for strong test performance. Details of the hybrid strategy are provided in **Appendix A.2**.

Infill Criterion In designing the infill criteria, we draw inspiration from the EIC-MSSAEA method (Wu et al. 2024). Specifically, we employ five different criteria: ND-A, two variants of ND- ΔPBI that respectively emphasize convergence and diversity, and two types of EPDI that focus on exploration and exploitation, respectively. Detailed definitions are provided in **Appendix A.3**.

Model the Evolution Search Procedure as an MDP

To enable dynamic control over the surrogate-assisted multi-objective optimization process, we model the search as a discrete-time, finite-horizon Markov Decision Process (MDP), defined as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where γ denotes discount factor. **State Space**. \mathcal{S} denotes the state space that reflects optimization status. At each decision step t , the state vector $s_t = [\mathbf{z}_t, \rho_t] \in \mathbb{R}^{2h+1}$ combines a landscape representation $\mathbf{z}_t \in \mathbb{R}^{2h}$ from a two-stage ELA module and $\rho_t = t/FE_{\max}$ is a normalized scalar indicating the proportion of the evaluation budget that has been consumed up to step t .

Action Space. Define action space $\mathcal{A} = \{a_1, a_2, \dots, a_6\}$, where a_1 represents regenerating new candidate solutions, while others trigger true evaluations of elite individuals, with each action corresponding to one of the five predefined infill criteria. **Transition Dynamics**. The state transition P can be expressed as a conditional probability $p(s_{t+1}|s_t, a_t)$, but this distribution is not explicitly available due to the black-box nature of the optimization process. Nevertheless, this does not hinder learning, as our model-free reinforcement learning approach relies only on sampled trajectories, without requiring knowledge of the transition probabilities.

Reward Function. The reward function $r(s_t, a_t)$ provides feedback based on the impact of the selected action on optimization progress. If the agent continues surrogate evolution ($a_t = a_1$), it receives a neutral reward of 0. When true evaluation is triggered ($a_t \in \{a_2, \dots, a_6\}$), the reward reflects whether the newly evaluated solutions improve the Pareto front. Formally:

$$r_t = \begin{cases} 0, & a_t = a_1 \\ 1.0 + \lambda \cdot \sum_{i=1}^k \frac{d_i}{d_{\text{ref}}^{(i)}}, & \text{if the front is improved} \\ -1.0, & \text{otherwise.} \end{cases} \quad (5)$$

Here, k denotes the number of newly selected candidate solutions for true evaluation. For each solution, d_i represents

the Manhattan distance to the closest point on the previous Pareto front (before improvement), and $d_{\text{ref}}^{(i)}$ is the Manhattan distance from that closest point to the origin. This normalization ensures consistency across tasks with different scales and enables a more stable learning signal.

We formulate the meta-level objective of DB-SAEA as a bi-level optimization problem. The lower level uses a surrogate-assisted optimizer \mathcal{O} , while the upper level learns a meta-policy π_{θ_π} to guide the search across tasks $\mathcal{T}_i \sim \mathcal{D}$. At each step t , the state s_t is constructed by combining the bi-space landscape representation $\Lambda_{\theta_A}(\mathcal{O}, \mathcal{T}_i)$, extracted via the ELA module of DB-SAEA, and a progress scalar. Based on this state, the meta-policy selects an action $a_t = \pi_{\theta_\pi}(s_t)$. The overall training objective is to maximize the expected performance across tasks:

$$J(\theta) = \mathbb{E}_{\mathcal{T} \sim \mathcal{D}} [R(\mathcal{O}, \pi_{\theta_\pi}, \mathcal{T})] \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \text{perf}(\mathcal{O}, a_t, \mathcal{T}_i), \quad (6)$$

where $\text{perf}(\cdot)$ is a task-specific performance metric, N is the total number of training tasks, and $R(\cdot)$ represents total reward accumulated by following the meta-policy.

Training Method

As formalized in Equation (6), our objective is to maximize the expected cumulative performance achieved by the collaboration of the surrogate-assisted optimizer \mathcal{O} and the meta-policy π_{θ_π} across multiple tasks $\mathcal{T}_i \sim \mathcal{D}$. To enable transferability across diverse optimization problems, DB-SAEA explicitly considers the cross-task transferability of the meta-policy. We adopt a **parallel sampling** and **centralized training** paradigm, implemented via an offline reinforcement learning framework based on Dueling DQN (Wang et al. 2016). Multiple multi-objective optimization environments (i.e., tasks) interact in parallel with the current meta-policy to generate heterogeneous state-action-reward trajectories. These experiences are aggregated into a centralized replay buffer and used to jointly train a meta-policy with shared parameters. This sampling-training loop is iterated until convergence. Notably, the bi-space ELA module is co-trained with the meta-policy during the reinforcement learning process, enabling tight integration between landscape-aware representation learning and adaptive decision-making.

To enable parallel sampling, we employ **Ray** (Moritz et al. 2018), an open-source framework for parallel processing in machine learning applications. With Ray, the sampling tasks can be distributed across multiple CPUs and GPUs, allowing simultaneous interaction with multiple environments.

Experimental Studies

In this section, we aim to address the following research questions: **RQ1**: How stable and convergent is the training process of the meta-policy across multiple optimization environments? **RQ2**: How well does the proposed approach transfer to new tasks with higher-dimensional problem settings? **RQ3**: What are the advantages of using TabPFN over

GP in the context of optimization? **RQ4:** What is the impact of controlling only the evolutionary algorithm or only the infill criterion, compared to the proposed dual-control strategy? Additional experimental results are provided in [Appendix B](#).

Experimental Setup

Benchmark Problems and Baselines To comprehensively validate the performance of DB-SAEA, experiments were conducted on nine benchmark problems, 2- and 3-objective ZDT1-3 (Zitzler, Deb, and Thiele 2000) and DTLZ2-7 (Deb et al. 2005). Moreover, we compare DB-SAEA against multiple state-of-the-art and classical algorithms, including CDM-PSL (Li et al. 2025), MOEA/DEGO (Zhang et al. 2009), USeMO-EI (Belakaria et al. 2020), qNEHVI (Daulton, Balandat, and Bakshy 2020), and NSGA-II (Deb et al. 2002). Additionally, we include an ablation variant, denoted as DB-SAEA-NSGA, in which only NSGA-III is used to generate candidate solutions, thereby removing the hybrid candidate generation strategy from the original DB-SAEA framework.

Training and Testing Settings We employ a parallel sampling strategy to interact with multiple benchmark environments simultaneously. All experiences are stored in a centralized replay buffer for joint training of the meta-policy and bi-space ELA module. The buffer is cleared and refreshed after each training round. The training batch size is set to 64. We use the Adam optimizer with an initial learning rate of 0.0001. To assess transferability, we apply a **leave-one-task-out cross-validation** strategy. In each run, we train the meta-policy and ELA module on eight tasks and then test them on the remaining unseen task, which has a higher problem dimensionality. This process is repeated 9 times, ensuring that each task is used once as the test task. Moreover, each test round is independently repeated 10 times. During both training and testing phases, we initialize the population in the true evaluation space with 80 solutions, while setting the remaining evaluation budget to 40 for use during the optimization process. At each decision step, one solution is selected for true evaluation. All attention modules within the bi-space ELA architecture employ a single-head configuration with hidden dimension $h = 16$.

All experiments were conducted on a compute cluster equipped with $4 \times 48\text{GB}$ VGPUs and $4 \times 25\text{-core Intel Xeon Platinum 8481C}$ processors. With the Ray framework, we achieved efficient task-level parallel sampling and centralized training, reducing each sampling-training cycle to around 5 minutes.

Evaluation Metric We employ the hypervolume (HV) metric (Zitzler and Thiele 2002) to evaluate the quality of the obtained solution sets. A higher HV value reflects better performance.

Training Stability and Convergence (RQ1)

We conduct 9 training runs using a leave-one-task-out cross-validation strategy. In each run, eight optimization environments are used for training, and each environment is configured with three different dimensional settings (15, 20, and

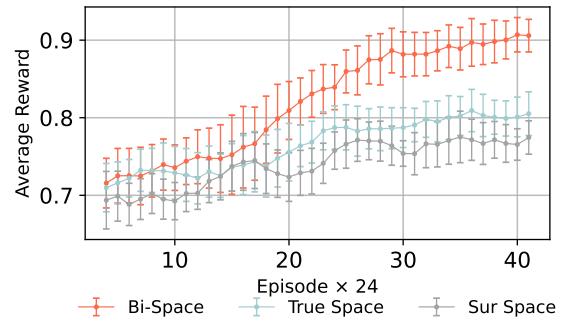


Figure 3: Average reward per true evaluation

25), resulting in a total of 24 training tasks. Figure 3 presents the average reward per true evaluation, obtained by averaging across environments at each training episode. The curve is smoothed using a moving average with a window size of 5. We compare three ELA variants during training: Bi-Space (features from both true and surrogate evaluations), True Space (only true-evaluation features), and Sur Space (only surrogate-based features). Bi-Space achieves higher rewards, demonstrating that combining accurate but sparse true data with low-cost surrogate information enhances optimization guidance. Vertical bars indicate ± 1 standard deviation over 9 runs.

Zero-shot Performance (RQ2)

To assess zero-shot transferability, we evaluate DB-SAEA on unseen tasks with higher dimensionality. After training on eight benchmark tasks, the model is directly tested on a held-out task with 30 decision variables. Figure 4 shows that DB-SAEA generally outperforms all baseline methods. In particular, DB-SAEA significantly outperforms its ablated variant DB-SAEA-NSGA, validating the benefit of the hybrid candidate generation strategy. Notably, although DB-SAEA adopts candidate generation components from CDM-PSL and qNEHVI, the incorporation of dual control yields a substantial performance gain beyond these individual components. While DB-SAEA performs comparably to CDM-PSL on simpler problems like ZDT1 and ZDT2, it shows marked improvements on more complex tasks.

Comparing TabPFN with Gaussian Process (RQ3)

Figure 5 compares the predictive performance of TabPFN and GP on candidate solutions selected for true evaluation across nine 30-dimensional benchmark tasks. At each true evaluation step, the figure shows the mean predicted values across all objectives, along with the corresponding mean true objective values for the selected solutions. The blue line represents the multi-objective mean prediction produced by TabPFN, the orange line indicates that of the GP model, and the gray line shows the actual mean of the true objective values. Shaded areas denote the ± 1 standard deviation around each model's predictions, reflecting prediction uncertainty. It can be observed that TabPFN yields predictions that are overall closer to the true values, with most true values falling

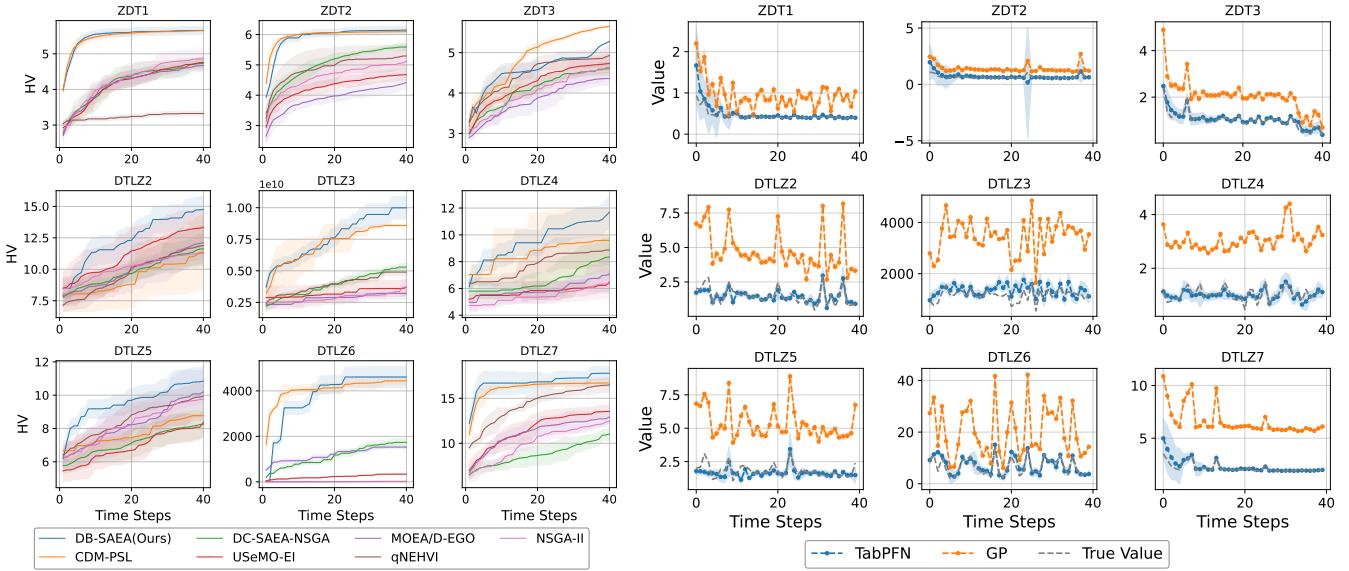


Figure 4: Performance comparison on unseen 30D tasks

Figure 5: Prediction and uncertainty of surrogate models

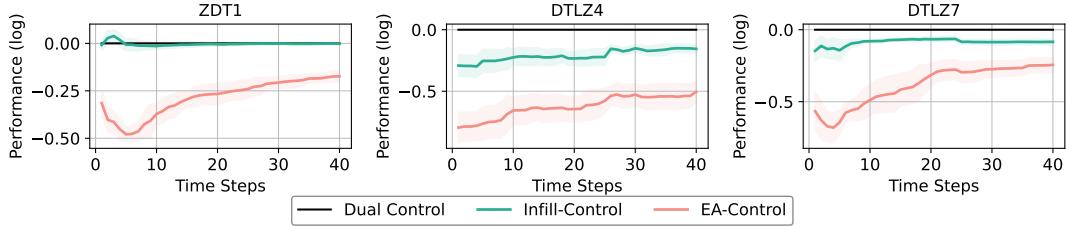


Figure 6: Performance comparison of single and dual control

within its ± 1 std range, whereas the GP model displays more noticeable deviations.

Comparison of Single and Dual Control (RQ4)

In this experiment, we construct two ablation variants to evaluate the effectiveness of the proposed dual-control mechanism of DB-SAEA: Infill Control, which only governs the selection of infill criteria, and EA Control, which only decides whether to perform true evaluation or continue surrogate-assisted search. Figure 6 presents a performance comparison across control strategies using the logarithmic hypervolume ratio computed with base 2. DB-SAEA with dual control achieves notably better performance on complex tasks such as DTLZ4 and DTLZ7, while showing comparable results to Infill Control on simpler problems like ZDT1. Evidently, Infill Control outperforms EA Control, suggesting that dynamic control over infill criterion selection contributes more significantly to optimization performance.

Conclusion

In this paper, we propose DB-SAEA, a MetaBBO framework tailored for expensive multi-objective optimization

problems, which learns a meta-policy to jointly control evolutionary algorithm and infill criterion selection. It integrates a bi-space ELA module to capture optimization states from both true and surrogate evaluation spaces, while ensuring scalability across problem dimensions, population sizes, and objectives. In addition, we employ TabPFN as a surrogate model to provide accurate predictions and uncertainty estimates without requiring task-specific training. Experiments show that DB-SAEA achieves strong performance, particularly on complex tasks, and demonstrates effective zero-shot transfer capability. Ablation studies validate the effectiveness of both the dual-control mechanism and the bi-space landscape modeling. Moreover, TabPFN significantly outperforms Gaussian Processes across all benchmark tasks. Future work may explore three directions: 1) expanding beyond dual control to incorporate more optimization decisions, such as surrogate model selection or evolutionary algorithm configuration, to enhance adaptive capability; 2) enhancing the joint modeling of true and surrogate landscape features by integrating them through advanced mechanisms such as cross-attention; and 3) further improving DB-SAEA for more complex scenarios, such as constrained multi-objective or dynamic optimization problems, to advance meta-level control in BBO.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant No. 72301286, 72431011).

References

- Akbari, H.; and Kazerooni, A. 2020. KASRA: A Kriging-based Adaptive Space Reduction Algorithm for global optimization of computationally expensive black-box constrained problems. *Applied Soft Computing*, 90: 106154.
- Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; and Koyama, M. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2623–2631.
- Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.
- Belakaria, S.; Deshwal, A.; Jayakodi, N. K.; and Doppa, J. R. 2020. Uncertainty-aware search framework for multi-objective Bayesian optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 10044–10052.
- Cai, X.; Gao, L.; and Li, X. 2019. Efficient generalized surrogate-assisted evolutionary algorithm for high-dimensional expensive problems. *IEEE Transactions on Evolutionary Computation*, 24(2): 365–379.
- Daulton, S.; Balandat, M.; and Bakshy, E. 2020. Differentiable Expected Hypervolume Improvement for Parallel Multi-Objective Bayesian Optimization. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 9851–9864. Curran Associates, Inc.
- Deb, K.; and Jain, H. 2013. An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part I: solving problems with box constraints. *IEEE transactions on evolutionary computation*, 18(4): 577–601.
- Deb, K.; Pratap, A.; Agarwal, S.; and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2): 182–197.
- Deb, K.; Thiele, L.; Laumanns, M.; and Zitzler, E. 2005. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization: theoretical advances and applications*, 105–145. Springer.
- Guo, H.; Ma, Z.; Chen, J.; Ma, Y.; Cao, Z.; Zhang, X.; and Gong, Y.-J. 2025. Configx: Modular configuration for evolutionary algorithms via multitask reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 26982–26990.
- He, C.; Zhang, Y.; Gong, D.; and Ji, X. 2023. A review of surrogate-assisted evolutionary algorithms for expensive optimization problems. *Expert Systems with Applications*, 217: 119495.
- Hollmann, N.; Müller, S.; Eggensperger, K.; and Hutter, F. 2023. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second. In *The Eleventh International Conference on Learning Representations*.
- Hollmann, N.; Müller, S.; Purucker, L.; Krishnakumar, A.; Körfer, M.; Hoo, S. B.; Schirrmeister, R. T.; and Hutter, F. 2025. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045): 319–326.
- Huixiang, Z.; Wenyin, G.; and Ling, W. 2021. Data-driven evolutionary sampling optimization forexpensive problems. *Journal of Systems Engineering and Electronics*, 32(2): 318–330.
- Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 448–456. pmlr.
- Jin, Y. 2011. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2): 61–70.
- Lange, R.; Schaul, T.; Chen, Y.; Lu, C.; Zahavy, T.; Dalibard, V.; and Flennerhag, S. 2023. Discovering Attention-Based Genetic Algorithms via Meta-Black-Box Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’23, 929–937. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701191.
- Li, B.; Di, Z.; Lu, Y.; Qian, H.; Wang, F.; Yang, P.; Tang, K.; and Zhou, A. 2025. Expensive multi-objective bayesian optimization based on diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 27063–27071.
- Lim, D.; Jin, Y.; Ong, Y.-S.; and Sendhoff, B. 2009. Generalizing surrogate-assisted evolutionary computation. *IEEE Transactions on Evolutionary Computation*, 14(3): 329–355.
- Liu, Q.; Wu, X.; Lin, Q.; Ji, J.; and Wong, K.-C. 2021. A novel surrogate-assisted evolutionary algorithm with an uncertainty grouping based infill criterion. *Swarm and Evolutionary Computation*, 60: 100787.
- Liu, Y.; and Wang, H. 2023. Surrogate-assisted hybrid evolutionary algorithm with local estimation of distribution for expensive mixed-variable optimization problems. *Applied Soft Computing*, 133: 10957.
- Ma, Z.; Chen, J.; Guo, H.; and Gong, Y.-J. 2025a. Neural Exploratory Landscape Analysis for Meta-Black-Box-Optimization. In *The Thirteenth International Conference on Learning Representations*.
- Ma, Z.; Guo, H.; Chen, J.; Li, Z.; Peng, G.; Gong, Y.-J.; Ma, Y.; and Cao, Z. 2023. MetaBox: A Benchmark Platform for Meta-Black-Box Optimization with Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 36, 10775–10795.
- Ma, Z.; Guo, H.; Gong, Y.-J.; Zhang, J.; and Tan, K. C. 2025b. Toward automated algorithm design: A survey and practical guide to meta-black-box-optimization. *IEEE Transactions on Evolutionary Computation*.
- Mersmann, O.; Bischi, B.; Trautmann, H.; Preuss, M.; Weihns, C.; and Rudolph, G. 2011. Exploratory landscape analysis. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 829–836.

- Moritz, P.; Nishihara, R.; Wang, S.; Tumanov, A.; Liaw, R.; Liang, E.; Elibol, M.; Yang, Z.; Paul, W.; Jordan, M. I.; et al. 2018. Ray: A distributed framework for emerging {AI} applications. In *13th USENIX symposium on operating systems design and implementation (OSDI 18)*, 561–577.
- Müller, S.; Hollmann, N.; Arango, S. P.; Grabocka, J.; and Hutter, F. 2022. Transformers Can Do Bayesian Inference. In *International Conference on Learning Representations*.
- Ning, W.; Guo, B.; Guo, X.; Li, C.; and Yan, Y. 2018. Reinforcement learning aided parameter control in multi-objective evolutionary algorithm based on decomposition. *Progress in Artificial Intelligence*, 7(4): 385–398.
- Noack, M. M.; Zwart, P. H.; Ushizima, D. M.; Fukuto, M.; Yager, K. G.; Elbert, K. C.; Murray, C. B.; Stein, A.; Doerr, G. S.; Tsai, E. H.; et al. 2021. Gaussian processes for autonomous data acquisition at large-scale synchrotron and neutron facilities. *Nature Reviews Physics*, 3(10): 685–697.
- Renau, Q.; Doerr, C.; Dreo, J.; and Doerr, B. 2020. Exploratory landscape analysis is strongly sensitive to the sampling strategy. In *International Conference on Parallel Problem Solving from Nature*, 139–153. Springer.
- Renau, Q.; Dreo, J.; Doerr, C.; and Doerr, B. 2019. Expressiveness and robustness of landscape features. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2048–2051.
- Seeger, M. 2004. Gaussian processes for machine learning. *International journal of neural systems*, 14(02): 69–106.
- Seiler, M. V.; Kerschke, P.; and Trautmann, H. 2025. Deep-ela: Deep exploratory landscape analysis with self-supervised pretrained transformers for single-and multi-objective continuous optimization problems. *Evolutionary Computation*, 1–27.
- Shao, S.; Tian, Y.; and Zhang, Y. 2025. Deep reinforcement learning assisted surrogate model management for expensive constrained multi-objective optimization. *Swarm and Evolutionary Computation*, 92: 101817.
- Sun, J.; Liu, X.; Bäck, T.; and Xu, Z. 2021. Learning adaptive differential evolution algorithm from optimization experiences by policy gradient. *IEEE Transactions on Evolutionary Computation*, 25: 666–680.
- Tahernezhad-Javazm, F.; Rankin, D.; Bois, N. D.; Smith, A. E.; and Coyle, D. 2024. R2 indicator and deep reinforcement learning enhanced adaptive multi-objective evolutionary algorithm. *arXiv preprint arXiv:2404.08161*.
- Tsaban, T.; Varga, J. K.; Avraham, O.; Ben-Aharon, Z.; Khramushin, A.; and Schueler-Furman, O. 2022. Harnessing protein folding neural networks for peptide–protein docking. *Nature communications*, 13(1): 176.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30.
- Wang, Z.; Schaul, T.; Hessel, M.; Hasselt, H.; Lanctot, M.; and Freitas, N. 2016. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, 1995–2003. PMLR.
- Wu, D.; and Wang, G. G. 2022. Employing reinforcement learning to enhance particle swarm optimization methods. *Engineering Optimization*, 54(2): 329–348.
- Wu, H.; Chen, Q.; Chen, J.; Jin, Y.; Ding, J.; Zhang, X.; and Chai, T. 2024. A multi-stage expensive constrained multi-objective optimization algorithm based on ensemble infill criterion. *IEEE Transactions on Evolutionary Computation*.
- Yang, X.; Wang, R.; Li, K.; and Ishibuchi, H. 2025. Meta-Black-Box optimization for evolutionary algorithms: Review and perspective. *Swarm and Evolutionary Computation*, 93: 101838.
- Zhan, D.; and Xing, H. 2021. A fast kriging-assisted evolutionary algorithm based on incremental learning. *IEEE transactions on evolutionary computation*, 25(5): 941–955.
- Zhang, Q.; Liu, W.; Tsang, E.; and Virginas, B. 2009. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3): 456–474.
- Zhang, Z.; Chen, H. C.; and Cheng, Q. S. 2020. Surrogate-assisted quasi-Newton enhanced global optimization of antennas based on a heuristic hypersphere sampling. *IEEE Transactions on Antennas and Propagation*, 69(5): 2993–2998.
- Zitzler, E.; Deb, K.; and Thiele, L. 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2): 173–195.
- Zitzler, E.; and Thiele, L. 2002. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4): 257–271.