Appendix of "Relation-Augmented Dueling Bayesian Optimization via Preference Propagation"

Xiang Xia[1] , **Xiang Shu**[1] , **Shuo Liu**[1,2] , **Yiyi Zhu**[1] , **Yijie Zhou**[1] , **Weiye Wang**[1] ,
Bingdong Li[1] and **Hong Qian**[1*]
[1]Shanghai Institute of AI Education, and School of Computer Science and Technology,
East China Normal University, Shanghai 200062, China
[2]Game AI Center, Tencent Inc, Shenzhen 518057, China
{10225102442, 51255901138}@stu.ecnu.edu.cn, seokliu@tencent.com, {51265901040,
10235102524, 10215102506}@stu.ecnu.edu.cn, {bdli, hqian}@cs.ecnu.edu.cn

This appendix first introduces the background of dueling Bayesian optimization. Then, we provide the notation used in the proposed method, followed by the pseudo-code of the preference propagation technique. Next, we describe the implementation details of the optimization methods compared in our experiments. Subsequently, comprehensive experimental results are presented, along with an analysis of the hyperparameter settings.

Algorithm 1 Dueling Bayesian Optimization (DBO)

```
Input: Initial dataset $\mathcal{D}_M = \{[\boldsymbol{x}_i; \boldsymbol{x}'_i], p_i\}_{i=1}^{M}$, number of
    available duels $N$, boundary of subspace $\mathcal{X} \subset \mathbb{R}^D$.
Procedure:
 1: for $j = M$ to $M + N - 1$ do
 2:     Fit a $\mathcal{GP}$ to $\mathcal{D}_j$ and learn $\pi_{f_p,j}([\boldsymbol{x}; \boldsymbol{x}'])$.
 3:     Sample a function $\pi_{\hat{f}_p}$ from $\mathcal{GP}$.
 4:     $\boldsymbol{x}_{\text{next}} = \text{argmax}_{\boldsymbol{x} \in \mathcal{X}} \int_{\mathcal{X}} \pi_{\hat{f}_p}([\boldsymbol{x}; \boldsymbol{x}']; \mathcal{D}_j) \mathrm{d}\boldsymbol{x}'$ .
 5:     $\boldsymbol{x}'_{\text{next}} = \text{argmax}_{\boldsymbol{x}' \in \mathcal{X}} \sigma(\mathcal{GP} | \boldsymbol{x} = \boldsymbol{x}_{\text{next}}, \mathcal{D}_j)$ .
 6:     Run the duel $[\boldsymbol{x}_{\text{next}}; \boldsymbol{x}'_{\text{next}}]$ and obtain $p_{j+1}$.
 7:     Augment $\mathcal{D}_{j+1} = \{\mathcal{D}_j \cup ([\boldsymbol{x}_{\text{next}}; \boldsymbol{x}'_{\text{next}}], p_{j+1})\}$.
 8: end for
 9: Fit a $\mathcal{GP}$ to $\mathcal{D}_{M+N}$ and find the solution $\boldsymbol{x}^*$ with the
    highest soft-Copeland score.
10: return $\boldsymbol{x}^*$.
```

A Dueling Bayesian Optimization

Dueling Bayesian Optimization [González *et al.*, 2017] is a method for dueling optimization, designed to optimize solutions based on pairwise preferences instead of explicit function values. The algorithm begins by initializing a dataset $\mathcal{D}_M$, which contains M initial duels $[\boldsymbol{x}_i; \boldsymbol{x}'_i]$ and their corresponding preferences p_i, along with a predefined budget N for the number of duels and the boundary of the search space $\mathcal{X} \subset \mathbb{R}^D$. At each iteration, a Gaussian process (GP) is fitted to the current dataset $\mathcal{D}_j$ to model the posterior distribution of the preference function $\pi_{f_p,j}([\boldsymbol{x}; \boldsymbol{x}'])$, capturing both the mean and uncertainty estimates (line 2). Using this GP model, a preference function sample $\pi_{\hat{f}_p}$ is drawn to guide the selection of the next duel (line 3). The next solution $\boldsymbol{x}_{\text{next}}$ is determined by maximizing the integral of the soft-Copeland score over the search space $\mathcal{X}$ (line 4). Subsequently, the solution $\boldsymbol{x}'_{\text{next}}$ is selected by maximizing the uncertainty of the GP posterior, conditioned on $\boldsymbol{x}_{\text{next}}$, to ensure exploration (line 5). A duel is then conducted between $[\boldsymbol{x}_{\text{next}}; \boldsymbol{x}'_{\text{next}}]$, and the resulting preference p_{j+1} is obtained from the preference function (line 6). The dataset is augmented with the new duel and preference feedback (line 7), and the GP model is updated accordingly. This iterative process continues until the predefined budget N is exhausted. Finally, after all duels are completed, the GP model is fitted to the final dataset $\mathcal{D}_{M+N}$, and the solution with the highest soft-Copeland score is identified as the optimal solution $\boldsymbol{x}^*$ (line 9). The entire process is summarized in Algorithm 1.

Symbol	Meaning	Symbol	Meaning
$\mathcal{X}$	Solution space	$\mathcal{D}$	Dataset
$\boldsymbol{x}$	Solution	$[\boldsymbol{x}, \boldsymbol{x}']$	Duel
p	Preference	π_{f_p}	Preference function
$\mathcal{G}$	Directed hypergraph	$\mathcal{V}$	A set of vertices
ε	Directed hyperedge	$\mathcal{E}$	A set of directed hyperedges
k	Preference propagation parameter	$\mathcal{GP}$	Gaussian process
I	Number of iterations	M	Number of initial solutions
N	Number of duels	D	Dimension of the solution space
$\mathcal{K}$	Similarity Model		

Table 1: Notation for the proposed method.

B Notation for the Proposed Method

In order to facilitate a better understanding of the proposed method, we present the notation used throughout this paper. Table 1 summarizes the key symbols and their corresponding meanings, providing clarity on the mathematical components and variables involved in our approach, with all other symbols derived from those in the Table 1.

C Pseudo-Code of the Preference Propagation Technique

The preference propagation technique, as shown in Algorithm 2, is designed to make fuller utilization of the existing pairwise preferences by modeling potential dueling rela-

[*] Corresponding Author.

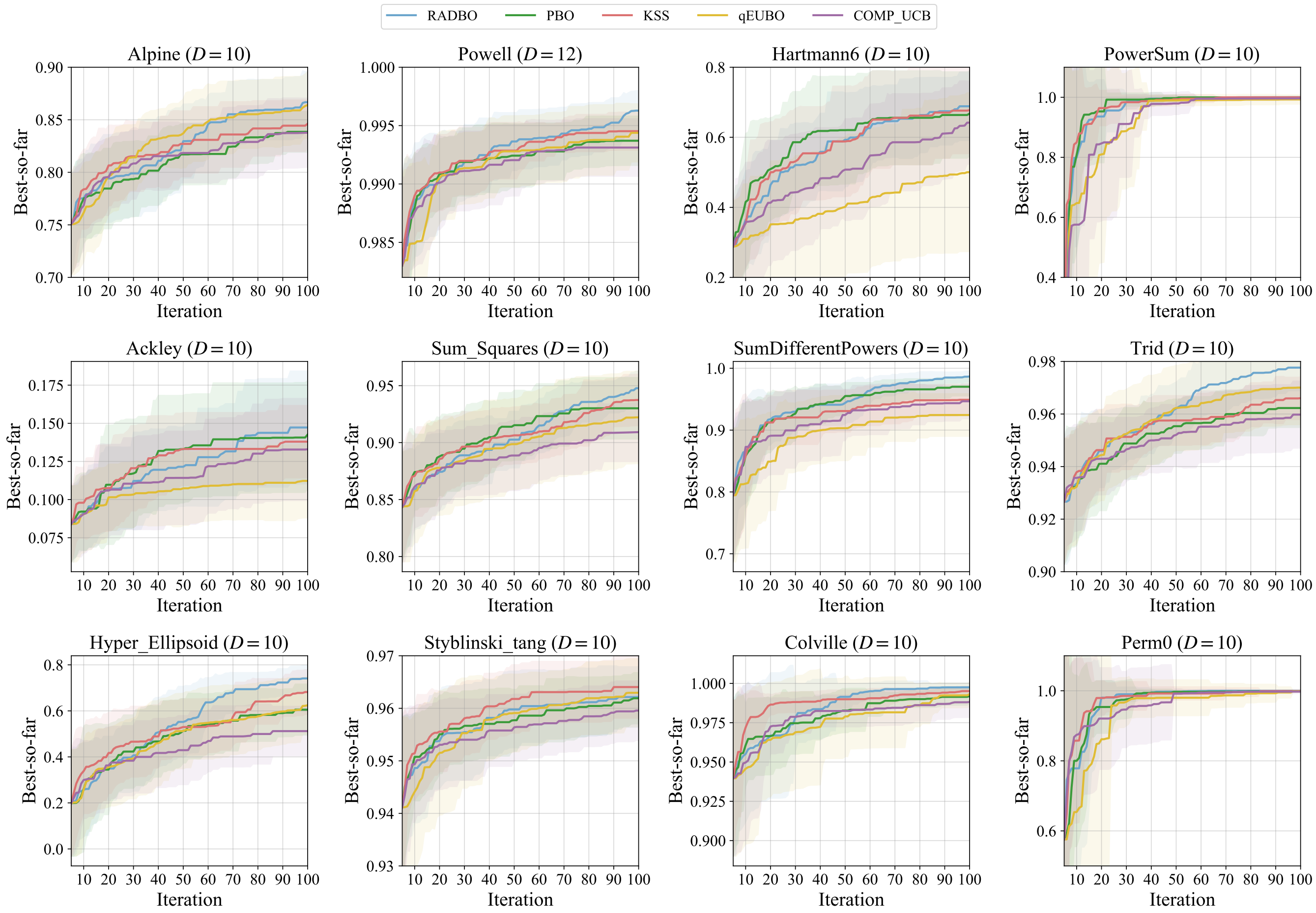

Figure 1: The best function value found by RADBO on the remaining synthetic functions are compared with different dueling optimization algorithms. All methods are evaluated with 5 initial duels, 95 iterations, and each experiment is repeated 20 times. The mean and standard deviation of the results are plotted. The horizontal axis of the plots represents the number of evaluations, and the vertical axis represents the best function value found by the algorithm.

tions among solutions. Initially, it requires the current dataset $\mathcal{D}_j$ and a preference propagation parameter k. The preference propagation technique begins by fitting an adaptive RBF kernel-based similarity model $\mathcal{K}$ to the dataset $\mathcal{D}_j$ and computing the covariance between solutions to assess their similarity (line 1). Next, the technique calculates distances based on the complement of similarity and clusters the solutions into k sets (line 2). From these clusters, it identifies a set of similar solutions $\mathcal{V}_\text{s}$ with the highest intra-cluster similarity, as well as the set of bad solutions $\mathcal{V}_\text{bad}$ and the set of good solutions $\mathcal{V}_\text{good}$ (line 3). Directed hyperedges are constructed to model the potential dueling relations among these solution sets (line 4). Finally, the augmented dueling relations are combined with the original dataset $\mathcal{D}_j$ to create an augmented dataset $\mathcal{D}_j^+$ (line 5), which is then returned as output. This technique aims to better uncover and utilize the potential dueling relations between preferences, thereby making fuller utilization of the existing pairwise preferences.

Algorithm 2 Preference Propagation Technique

```
Input: Current dataset $\mathcal{D}_j = \{[\boldsymbol{x}_i; \boldsymbol{x}'_i], p_i\}_{i=1}^{j}$, and preference propagation parameter $k$.
Procedure:
1: Fit a similarity model $\mathcal{K}$ to $\mathcal{D}_j$ and compute the covariance between any two solutions in the dataset $\mathcal{D}_j$ to assess their similarity.
2: Cluster all solutions into $k$ sets based on the similarity.
3: Obtain the set of similar solutions $\mathcal{V}_\text{s}$ with the highest intra-cluster similarity, the set of bad solutions $\mathcal{V}_\text{bad}$ and the set of good solutions $\mathcal{V}_\text{good}$.
4: Construct the directed hyperedges $\varepsilon_1$ and $\varepsilon_2$ to model the potential dueling relations.
5: Combine the augmented dueling relations with the dataset $\mathcal{D}_j$ and obtain the augmented dataset $\mathcal{D}_j^+$.
6: return the augmented dataset $\mathcal{D}_j^+$.
```

D Implementation Details of the Compared Optimization Methods in Experiments

PBO [González *et al.*, 2017]: PBO, the first framework to extend Bayesian optimization to scenarios where only information about user preferences can be obtained, is repeated using the BoTorch framework in the experiments and follows the same hyper-parameter specifications as outlined in Zhang *et al.* [2023].

KSS [Sui *et al.*, 2017]: KSS is an algorithm that effectively addresses the multi-dueling bandits problem by reducing it to a conventional bandit setting, and it can also be applied to dueling optimization. We use the code from the GitHub repository: https://github.com/Zhangywh/PE-DBO.

COMP-UCB [Xu *et al.*, 2020]: COMP-UCB is the simplified version that omits the second part of the optimization process that depends on function values. We use the code from the GitHub repository: https://github.com/Zhangywh/PE-DBO.

qEUBO [Astudillo *et al.*, 2023]: qEUBO provides a promising acquisition function with a grounded decision-theoretic justification. We use the implementation from the author's GitHub repository: https://github.com/RaulAstudillo06/qEUBO.

GP-UCB [Srinivas *et al.*, 2010]: GP-UCB is a Bayesian optimization algorithm with the upper confidence bound strategy that builds a model to predict an unknown function, balancing exploration and exploitation. In the experiments, the BoTorch framework is used to implement GP-UCB, with β defined as $0.2D\log(2n)$, where D is the dimension of the solution space and n is the number of samples in the dataset.

GP-EI [Garnett, 2023]: GP-EI is a Bayesian optimization algorithm with the expected improvement strategy that builds a model to predict an unknown function, balancing exploration and exploitation. In the experiments, the BoTorch framework is used to implement GP-EI with default hyper-parameters.

E Detailed Experimental Results

In the synthetic functions and real-world tasks experiments, $I = 500$ samples are employed to estimate the integral of the soft-Copeland score, and the GP model is initialized using $M = 5$ duels, followed by $N = 95$ duels for the optimization process. For RADBO, we use $k = 3$ to execute the preference propagation technique. Figure 1 and Figure 2 present the experimental results on the remaining 12 synthetic functions and 2 real-world datasets. For synthetic functions, those with fewer than 10 dimensions (e.g., Hartmann6) are extended to 10 dimensions, while functions with more than 10 dimensions (e.g., Powell) retain their original dimensionality. All experiments on synthetic functions and real-world tasks are maximization optimization.

For all tasks, the best function value found so far is used as the evaluation criterion, with min-max scaling applied to the values to enable a more intuitive analysis of the results. For synthetic functions or real datasets where the optimal function value is unknown, we use grid sampling of a large number of solutions to approximate the optimum. It is worth noting that, in all tasks, function values are only used for presenting the final results and are not utilized during the dueling optimization process.

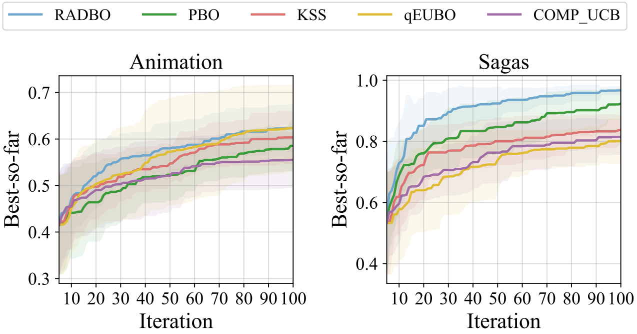

Figure 2: The best function value found by algorithms on the remaining two real-world datasets. Each experiment is repeated 20 times. The mean and standard deviation of the results are plotted. The horizontal axis of the plots represents the number of evaluations, and the vertical axis represents the best function value found by the algorithm. All methods are evaluated with 5 initial duels and 95 iterations.

As shown in Figure 1 and Figure 2, in most experimental scenarios, RADBO achieves the best optimization performance, verifying its capability to effectively handle dueling optimization tasks and highlighting its superiority over other dueling optimization methods. Moreover, compared to PBO (a version of RADBO without the preference propagation technique), RADBO achieves better optimization performance, indicating that the preference propagation technique indeed enhances the optimization performance of algorithm.

Table 2 and Table 3 record the final mean convergence value of various algorithms under each experimental environment. In order to verify that RADBO statistically outperforms other baselines in most cases, we perform t-tests with a significance level of 0.1. As shown in the tables, in most tasks, RADBO statistically outperforms other dueling optimization methods. The results show that RADBO can handle dueling optimization tasks well and reflect the superiority of RADBO over other dueling optimization methods.

F Hyper-Parameter Analysis

The hyper-parameter analysis experiments for k are conducted on all synthetic functions, with the results shown in Figure 3. In the experiments, $I = 500$ samples are employed to estimate the integral of the soft-Copeland score, and the GP model is initialized using $M = 5$ duels, followed by $N = 95$ duels for the optimization process. For RADBO, a series of hyper-parameter values for k are used to execute the preference propagation technique. For comparison, the final results of PBO, regarded as a version of RADBO after ablating the preference propagation technique, are also plotted. The results clearly show that RADBO consistently surpasses PBO across various hyper-parameter values of k, highlighting its insensitivity to changes in k. This characteristic ensures the adaptability and stability of RADBO across dif-

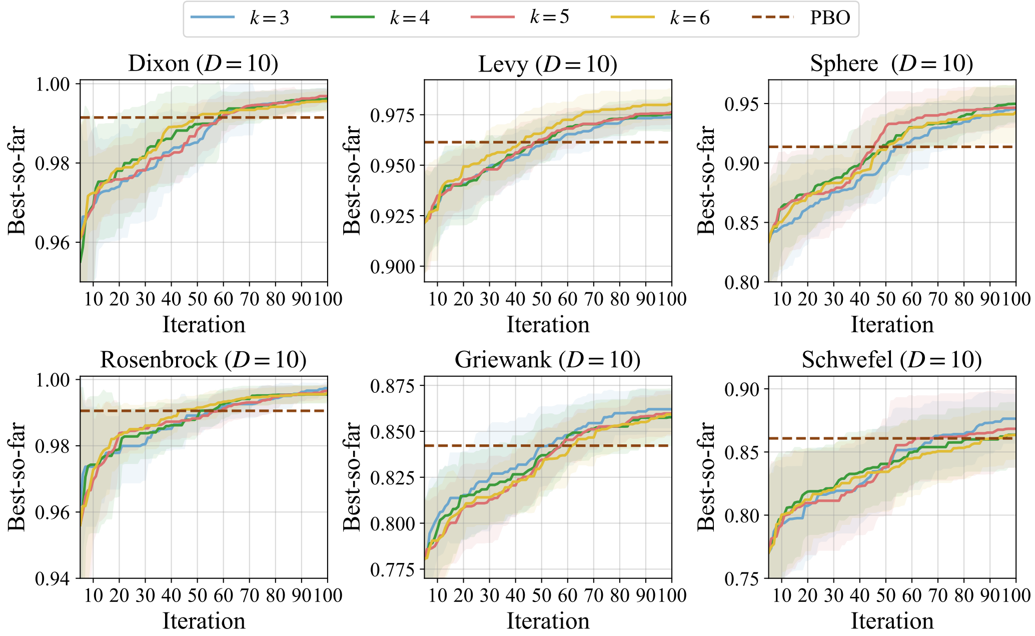

Figure 3: Hyper-parameter analysis on synthetic functions. Each experiment is repeated 20 times and the final results of PBO are also plotted. The mean and standard deviation of the best function value found are plotted. The horizontal axis of the plots represents the number of evaluations, and the vertical axis represents the best function value found by the algorithm.

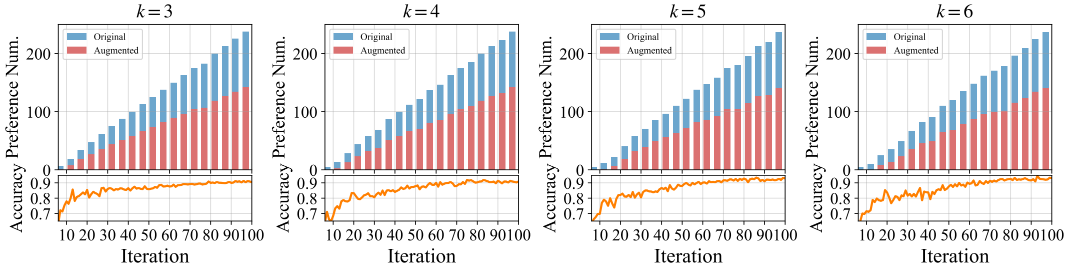

Figure 4: Investigating the behavior of the preference propagation technique under different hyper-parameter values of k on the Griewank function ($D = 10$). The figure shows the mean number of preferences in the original dataset (blue) and the augmented preferences newly added after preference propagation (red) in the top plot, as well as the mean accuracy of the augmented preferences in the bottom plot. All settings are the same as Figure 3, and each experiment is repeated 20 times.

ferent application scenarios. Furthermore, regardless of the hyper-parameter k, RADBO consistently shows a stable improvements in performance, which indicates that the preference propagation technique operates reliably, showcasing its reliability and consistency under varying conditions.

To further analyze why RADBO is not sensitive to changes in the hyper-parameter k, we explore the behavior of the preference propagation technique under different values of k on the Griewank function. Figure 4 shows the mean number of original preferences at the beginning of each iteration and the newly added augmented preferences after preference propagation (top), as well as the mean accuracy of the augmented preferences (bottom). From the figure, we observe that within a limited range, the choice of the hyper-parameter k does not significantly affect the number of new augmented preferences added after preference propagation, nor their accuracy during the optimization process. Therefore, the hyper-parameter k do not significantly affect the relation augmentation effect of the preference propagation technique on the existing dataset, allowing RADBO to achieve better optimization performance. In fact, within the preference propagation technique, after GMM performs clustering, we only select the solutions from the cluster with the highest intra-cluster similarity as the similar solutions, and the role of GMM is to help us select the most similar batch of solutions. Therefore, the choice of k does not significantly affect the performance of RADBO.

These results explain why the optimization performance of RADBO is not sensitive to changes in the hyper-parameter k and it further verifies that the preference propagation technique can run stably under different conditions.

Method	Rosenbrock	Dixon	Griewank	Levy	Schwefel	Sphere
PBO	0.991 ± 0.005	0.991 ± 0.005	$\underline{0.842 \pm 0.018}$	0.961 ± 0.010	$\underline{0.861 \pm 0.032}$	0.914 ± 0.021
KSS	$\underline{0.994 \pm 0.004}$	$\underline{0.992 \pm 0.004}$	0.838 ± 0.017	0.967 ± 0.010	0.858 ± 0.024	$\underline{0.930 \pm 0.020}$
qEUBO	0.990 ± 0.007	0.992 ± 0.005	0.818 ± 0.015	$\underline{0.970 \pm 0.012}$	0.850 ± 0.051	0.918 ± 0.020
COMP-UCB	0.988 ± 0.007	0.989 ± 0.004	0.825 ± 0.017	0.962 ± 0.009	0.843 ± 0.026	0.898 ± 0.025
RADBO	$\mathbf{0.997 \pm 0.002}$*	$\mathbf{0.996 \pm 0.003}$*	$\mathbf{0.862 \pm 0.010}$*	$\mathbf{0.974 \pm 0.006}$	$\mathbf{0.876 \pm 0.019}$*	$\mathbf{0.945 \pm 0.012}$*

Method	Alpine	Powell	Hartmann6	PowerSum	Ackley	Sum_Squares
PBO	0.839 ± 0.019	0.994 ± 0.002	0.668 ± 0.121	1.000 ± 0.001	$\underline{0.143 \pm 0.035}$	0.930 ± 0.027
KSS	0.846 ± 0.022	$\underline{0.995 \pm 0.002}$	$\underline{0.678 \pm 0.122}$	$\underline{1.000 \pm 0.000}$	0.138 ± 0.024	$\underline{0.937 \pm 0.023}$
qEUBO	$\underline{0.864 \pm 0.032}$	0.994 ± 0.003	0.500 ± 0.227	0.993 ± 0.016	0.112 ± 0.024	0.922 ± 0.041
COMP-UCB	0.838 ± 0.033	0.993 ± 0.003	0.642 ± 0.119	0.996 ± 0.007	0.133 ± 0.034	0.909 ± 0.026
RADBO	$\mathbf{0.867 \pm 0.029}$	$\mathbf{0.996 \pm 0.002}$*	$\mathbf{0.689 \pm 0.098}$	$\mathbf{1.000 \pm 0.000}$*	$\mathbf{0.147 \pm 0.037}$	$\mathbf{0.948 \pm 0.015}$

Method	SumDifferentPowers	Trid	Hyper_Ellipsoid	Styblinski_tang	Colville	Perm0
PBO	$\underline{0.970 \pm 0.0015}$	0.962 ± 0.007	0.605 ± 0.111	0.962 ± 0.005	0.992 ± 0.005	0.999 ± 0.002
KSS	0.949 ± 0.024	0.966 ± 0.008	$\underline{0.682 \pm 0.097}$	$\mathbf{0.964 \pm 0.006}$	$\underline{0.995 \pm 0.003}$	$\underline{0.999 \pm 0.001}$
qEUBO	0.924 ± 0.035	$\underline{0.970 \pm 0.014}$	0.625 ± 0.125	0.963 ± 0.006	0.993 ± 0.008	0.999 ± 0.003
COMP-UCB	0.947 ± 0.028	0.960 ± 0.013	0.512 ± 0.108	0.960 ± 0.006	0.988 ± 0.010	0.998 ± 0.003
RADBO	$\mathbf{0.987 \pm 0.009}$*	$\mathbf{0.978 \pm 0.009}$*	$\mathbf{0.741 \pm 0.057}$*	$\underline{0.962 \pm 0.006}$	$\mathbf{0.998 \pm 0.003}$*	$\mathbf{1.000 \pm 0.000}$*

Table 2: The detailed results of dueling optimization methods on synthetic functions, scaled using min-max normalization. In each column, an entry with the best mean value is marked in bold and underline for the runner-up. If the mean value of the best method significantly differs from the runner-up, passing a t-test with a significance level of 0.1, then we denote it with "*" at the corresponding position.

Method	RobotPush	Cassini1-MINLP	Carcab	Sagas	Animation
PBO	0.353 ± 0.111	$\underline{0.964 \pm 0.007}$	$\underline{0.956 \pm 0.010}$	$\underline{0.923 \pm 0.050}$	0.585 ± 0.054
KSS	0.355 ± 0.091	0.956 ± 0.017	0.953 ± 0.010	0.837 ± 0.046	0.603 ± 0.056
qEUBO	0.291 ± 0.094	0.947 ± 0.017	0.931 ± 0.026	0.801 ± 0.073	$\underline{0.623 \pm 0.092}$
COMP-UCB	$\underline{0.356 \pm 0.113}$	0.948 ± 0.018	0.943 ± 0.022	0.814 ± 0.056	0.555 ± 0.058
RADBO	$\mathbf{0.391 \pm 0.110}$	$\mathbf{0.971 \pm 0.011}$*	$\mathbf{0.962 \pm 0.011}$*	$\mathbf{0.967 \pm 0.012}$*	$\mathbf{0.624 \pm 0.050}$

Table 3: The detailed results of dueling optimization methods on real-world datasets, scaled using min-max normalization. In each column, an entry with the best mean value is marked in bold and underline for the runner-up. If the mean value of the best method significantly differs from the runner-up, passing a t-test with a significance level of 0.1, then we denote it with "*" at the corresponding position.

References

[Astudillo *et al.*, 2023] Raul Astudillo, Zhiyuan (Jerry) Lin, Eytan Bakshy, and Peter I. Frazier. qeubo: A decision-theoretic acquisition function for preferential Bayesian optimization. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206, pages 1093–1114, Valencia, Spain, 2023.

[Garnett, 2023] Roman Garnett. *Bayesian Optimization*. Cambridge University Press, Cambridge, England, 2023.

[González *et al.*, 2017] Javier González, Zhenwen Dai, Andreas C. Damianou, and Neil D. Lawrence. Preferential Bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1282–1291, Sydney, Australia, 2017.

[Srinivas *et al.*, 2010] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1015–1022, Haifa, Israel, 2010.

[Sui *et al.*, 2017] Yanan Sui, Vincent Zhuang, Joel W. Burdick, and Yisong Yue. Multi-dueling bandits with dependent arms. In *Proceedings of the 33th Conference on Uncertainty in Artificial Intelligence*, Sydney, Australia, 2017.

[Xu *et al.*, 2020] Yichong Xu, Aparna Joshi, Aarti Singh, and Artur Dubrawski. Zeroth order non-convex optimization with dueling-choice bandits. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence*, volume 124 of *Proceedings of Machine Learning Research*, pages 899–908, Virtual Event, 2020.

[Zhang *et al.*, 2023] Yangwenhui Zhang, Hong Qian, Xiang Shu, and Aimin Zhou. High-dimensional dueling optimization with preference embedding. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, pages 11280–11288, Washington, DC, 2023.