Proceedings of the International Conference on Industrial Engineering and Operations Management

Publisher: IEOM Society International, USA DOI: 10.46254/NA11.20260237
Published: June 9, 2026

Building Trust in Black-box Optimization: A Comprehensive Framework for Explainability

Nazanin Nezami
University of Illinois Chicago
Chicago, Illinois, USA
nnezam2@uic.edu

Hadis Anahideh
University of Illinois Chicago
Chicago, Illinois, USA
hadis@uic.edu

Abstract

Optimizing costly black-box functions within a constrained evaluation budget presents significant challenges in many real-world applications. Surrogate optimization (SO) is a common resolution; however, the complexity of surrogate models and the sampling core (e.g., acquisition functions) often introduces proprietary elements, leading to a lack of explainability and transparency. While existing literature has primarily focused on enhancing convergence to global optima, the practical interpretation of newly proposed strategies remains underexplored, particularly in batch evaluation settings. In this paper, we propose Inclusive Explainability Metrics for Surrogate Optimization (IEMSO), a comprehensive set of model-agnostic metrics designed to enhance the explainability of SO approaches. Through these metrics, we provide both intermediate and post-hoc explanations, enabling practitioners to build trust before and after conducting expensive evaluations. We consider four primary categories of metrics, each targeting a specific aspect of the SO process: Sampling Core Metrics, Batch Properties Metrics, Optimization Process Metrics, and Feature Importance Metrics. Our experimental evaluations demonstrate the significant potential of the proposed metrics across different benchmarks.

Keywords

Black-box Optimization, Explainability Framework, Explainable Machine Learning

1. Introduction

Interpretability and reliability are foundational elements of any optimization approach (Carvalho, Pereira, and Cardoso 2019), serving as essential pillars upon which trust and acceptance are built. While explainability is a well-studied concept for machine learning (ML) models (Linardatos, Papastefanopoulos, and Kotsiantis 2020), the application of these principles to black-box optimization algorithms, such as Surrogate Optimization or Bayesian optimization (Rodemann et al. 2024), remains relatively unexplored. Black-box optimization (BBO) techniques aim to find the best parameter configurations or designs that optimize a target function, which is often unknown and costly to evaluate. The general idea is to sample the solution space strategically to discover promising areas.

The importance of BBO cannot be overstated. It is widely used in various fields such as engineering design (W. Chen, Chiu, and Fuge 2019), hyperparameter tuning for ML models (Turner et al. 2021), and scientific lab experiments (Angermueller et al. 2020). These optimization techniques are crucial when the objective function is complex, expensive to evaluate, or lacks an explicit analytical form. With an efficient exploration and exploitation of the solution space, BBO helps identify optimal solutions that would be otherwise infeasible to find using traditional methods.

Despite their pivotal role in ML and other applications, black-box optimizers (acquisition functions) are frequently perceived as opaque and inscrutable. This lack of transparency hinders user trust and limits their broader applicability. Consequently, improving our understanding and interpretation of these optimization approaches can significantly enhance their usability and reliability in practical settings.

Example: The "Robot Pushing" problem involves two robot hands working together to push objects toward a designated goal. The movements of the robot's hands are influenced by 14 parameters, including hand trajectory, location, rotation, velocity, and direction. This complexity requires rigorous experimentation and precise optimization to achieve the desired outcomes. In this context, batch evaluation is crucial because it allows for efficient usage of computational resources and enables parallel processing. Evaluating a large batch of sample points simultaneously demands a transparent and explainable selection process. Consider the scenario where a BBO algorithm selects a batch of 50 configurations for evaluation. Engineers need to understand the rationale behind the choice of each configuration in the batch. This understanding is necessary to build confidence and trust in the optimizer's decisions and the overall optimization outcomes. For instance, engineers need to know if the selected configurations explore a diverse range of trajectories and velocities, or if they exploit the previously successful parameters. A transparent explanation of these choices can demonstrate how the algorithm balances exploration and exploitation, ensuring that the optimization process is effective and reliable.

Recent research has highlighted the importance of explainability in BBO. For instance, Adachi et al. (2023) discussed that Bayesian optimization (BO) often struggles to gain user trust due to its opaque nature. They introduced the Collaborative and Explainable Bayesian Optimization (CoExBO) framework, which uses preference learning to incorporate human knowledge and Shapley values to quantify the contribution of each feature to the acquisition function. Similarly, Chakraborty, Seifert, and Wirth (2024) introduced TNTRules, a post-hoc, rule-based explainability method that employs multi-objective optimization to generate high-quality explanations for BO. TNTRules uses hierarchical agglomerative clustering (HAC) to uncover rules within the optimization space. It provides local explanations through visual graphs and actionable rules, and global explanations through ranked "IF-THEN" rules. These rules explain how the selected points relate to the optimization objectives and guide users in understanding the optimization process. Rodemann et al. (2024) developed the ShapleyBO framework, which utilizes Shapley values to interpret proposed parameter configurations in BO. ShapleyBO quantifies each parameter's contribution to the configurations suggested by a BO algorithm based on their impact on the acquisition function. Shapley values help explain how each parameter influences this balance, providing transparency in the selection process.

Gaps and Contribution: Despite recent advancements in enhancing the explainability of BBO, existing methods such as CoExBO, TNTRules, and ShapleyBO primarily focus on Bayesian optimization and do not generalize well to other surrogate optimization techniques, limiting their applicability. Additionally, they merely look at the contribution of features to the optimizer (often Gaussian Processes (GPs)), leaving a gap in comprehensive explainability for other types of surrogate models and optimization approaches. These limitations underscore the need for a more comprehensive approach to explainability in surrogate optimization. In response to these gaps, this paper proposes *Inclusive Explainability Metrics for Surrogate Optimization (IEMSO)* to enhance the explainability of black-box optimization methods. Our proposed metrics are designed to be model-agnostic and can be applied to any framework, providing detailed explanations of various steps in surrogate optimization to improve human understanding and trust in the optimization process.

Our perspective and key contributions differ from existing explainable Bayesian optimization (BO) papers in several ways. Although a trivial application of our explainability metrics is for human collaboration with surrogate optimization, the primary focus is on gaining user trust rather than incorporating human feedback or learning human preferences, as emphasized in (Adachi et al. 2023). Our metrics are applicable to any BO sampling strategy, not limited to specific acquisition function formats like those discussed in (Rodemann et al. 2024). Furthermore, instead of solely focusing on post-hoc explanations as in (Chakraborty, Seifert, and Wirth 2024), our proposed metrics offer both intermediate and post-hoc interpretations, aiding practitioners before and after performing expensive evaluations. Unlike (Chakraborty, Seifert, and Wirth 2024), which focuses on explaining the parameter space, we also address the explanation of single and batch BO proposals.

This paper introduces a wide range of explainability metrics applicable to different stages of surrogate optimization. Our proposed metrics are designed to be generalizable to any optimizers and surrogate models, thereby broadening the scope and utility of explainability in BBO. This comprehensive set of metrics ensures that the decision-making processes in surrogate optimization is explainable, enhancing their applicability across diverse optimization scenarios. Our metrics are divided into four main categories, each focusing on a different aspect of the surrogate optimization process; 1) Sampling Core Metrics Explain the reasoning and strategy behind the sampling process. 2) Batch Properties Metrics Detail the criteria and efficiency of the generated sample batches. 3) Optimization Process Metrics Provide a comprehensive overview of the entire optimization process, including convergence and performance. 4) Feature Importance Metrics Describe the importance of features for the black-box objective, surrogate model behavior, and exploration or exploitation objectives. This comprehensive set of metrics ensures that the decision-making processes in surrogate optimization is explainable, broadening the scope and utility of explainability in black-box optimization.

2. Related Work

Research in BBO has concentrated on managing the exploration-exploitation trade-off (Frazier 2018), traditionally in a sequential evaluation setting (Jones, Schonlau, and Welch 1998; Srinivas et al. 2009). However, the introduction of batch sampling has created a paradigm shift, as it is not only necessary for numerous experimental evaluation scenarios but also accelerates convergence, enhances exploration, and improves robustness by leveraging parallel computational resources (Desautels, Krause, and Burdick 2014; Balandat et al. 2020).

Explainable AI (XAI) is a growing field focused on making artificial intelligence systems more transparent and understandable to humans (Arrieta et al. 2020). Researchers have developed various techniques to elucidate the inner workings of complex models, ranging from local explanation methods such as LIME (Local Interpretable Model-agnostic Explanations) (Ribeiro, Singh, and Guestrin 2016) and SHAP (SHapley Additive exPlanations) (Lundberg and Lee 2017) to global interpretation strategies such as decision trees and rule-based systems (Linardatos, Papastefanopoulos, and Kotsiantis 2020; Sharma et al. 2024). However, research on explainability in black-box optimization is sparse. Only a few articles in the XAI literature have addressed the explainability of black-box optimizers. For example, Seitz (2022) proposes a gradient-based explainability for GPs as the primary surrogate model in BO to create uncertainty-aware feature rankings. Explainable Metaheuristic (Singh, Brownlee, and Cairns 2022) mines surrogate fitness models with the objective of identifying the variables that strongly influence solution quality for providing an explanation of the near-optimal solution. Recently, Adachi et al. (2023) introduced Collaborative and Explainable Bayesian Optimization (CoExBo) specifically for lithium-ion battery design. This framework integrates human expertise into BO through preference learning and utilizes Shapley values to explain its recommendations. CoExBo begins by aligning human knowledge with BO via preference learning, followed by proposing multiple options from which the user can choose based on additional Shapley value insights. In contrast, Shapley-assisted human-BO collaboration in (Rodemann et al. 2024) directly leverages Shapley values to align a single BO proposal with human input. Additionally, Chakraborty, Seifert, and Wirth (2024) recently presented TNTRules, a post-hoc rule-based explanation method for BO. This method identifies parameter subspaces for tuning using clustering algorithms, highlighting the advantages of XAI methods in collaborative BO settings.

3. Inclusive Explainability Metrics for Surrogate Optimization

In this section, we introduce the Inclusive Explainability Metrics for Surrogate Optimization (IEMSO) framework, designed to enhance the explainability and trustworthiness of SO approaches. The IEMSO framework is structured around four primary categories of model-agnostic metrics, each targeting a specific aspect of the SO process.

3.1. Sampling Core Metrics

This category of metrics focuses on providing explanations for individual sample points selected for expensive evaluation in each iteration. They offer insights into the logic behind the sampling process. Sample points can be explained based on their contribution to the selected subset (batch), their location in the solution space, and their contribution to SO objective (exploration-exploitation trade-off).

Point Coordinate Exploration (PCE) is a metric that quantifies the extent to which the selected sample points have covered the full range of each coordinate or feature within the predefined bounds.

Definition 1. *Given a set of n evaluated sample points $\mathcal{D} = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^n\}$, where each $\mathbf{x}^i$ is a d-dimensional vector and each coordinates x_j, $\forall j = 1,2, \dots, d$) is bounded by $[lb_j, ub_j]$, the PCE is defined as the normalized range covered*

by the sample points in that coordinate, $PCE_j = \frac{\max\left(x_j^1,\ldots,x_j^n\right)-\min\left(x_j^1,\ldots,x_j^n\right)}{ub_j-lb_j}$*. Hence, the average PCE is* $PCE(\mathcal{D}) = \frac{1}{d}\sum_{j=1}^{d} PCE_j$.

A PCE value close to 1 indicates that the sample points have sufficiently explored the range of each coordinate $[lb_j, ub_j]$. Conversely, a PCE value approaching 0 indicates that the sample points have explored only a limited fraction of each coordinate's range. This metric is simple to compute and offers a clear measure of coverage for each dimension. By normalizing exploration extent, PCE ensures consistency across different coordinates, helping to assess whether the sampling strategy is effectively exploring the search space which is a key factor in optimization.

Mean Distance from Prior Evaluations (MDPE) is a metric designed to quantify the dissimilarity (i.e., distance) of each individual sample point from the previously evaluated data points. In SO, sample points that exhibit greater distance from the previously evaluated points are predominantly important for the exploration objective, whereas points that are similar or in close proximity to the evaluated set are primarily selected for exploitation purposes.

Definition 2. *Given a set of* n *evaluated sample points* $\mathcal{D} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n\}$*, and the selected subset of candidate points* $\mathcal{S} = \{\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}_k\}$*, the MDPE is defined as the average pairwise Euclidean distance between each sample point* $\mathbf{z}^i \in \mathcal{S}$ *and the set of evaluated points* $\mathcal{D}$*,* $MDPE(\mathbf{z}^k) = \frac{1}{n}\sum_{i=1}^{n} \| \mathbf{z}^k - \mathbf{x}^i \|$*, where* $\| \mathbf{z}^k - \mathbf{x}^i \|$ *denotes the Euclidean distance between the points* $\mathbf{z}^k$ *and* $\mathbf{x}^i$.

The Mean Distance from Prior Evaluations (MDPE) metric offers several key advantages. Firstly, it is intuitive, providing a clear and understandable measure of how different or distant the new sample points are from the already evaluated points. This clarity aids in the easy interpretation of the sampling strategy's effectiveness. Secondly, MDPE effectively highlights the balance between exploration and exploitation, a crucial aspect of optimization tasks. Distant points indicate exploration, while points closer to previously evaluated ones indicate exploitation. Finally, MDPE offers a quantitative measure to assess the diversity and novelty of the selected sample points, enabling a rigorous evaluation of the optimization process.

Contribution to the Hypervolume of Exploration-Exploitation (CHEE) is a metric that quantifies the expected and actual impact of each sample point on exploration and exploitation objectives, both prior to and following the expensive evaluation.

Definition 3. *Consider a surrogate optimization approach characterized by an exploitation metric* $\mu(\mathbf{x})$ *and an exploration metric* $\sigma(\mathbf{x})$*, where* $\mathbf{x} \in \mathcal{X}$ *represents a point in the solution space* $\mathcal{X}$*. Let* $\hat{\mu}(\mathbf{x})$ *and* $\hat{\sigma}(\mathbf{x})$ *denote the surrogate model's predictions for the exploitation and exploration metrics, respectively. The estimated Pareto set* $\mathcal{P}'$ *constructed on the estimated exploration and exploitation metrics is defined as* $\mathcal{P}' = \{\mathbf{x} \in \mathcal{X} \mid \nexists \mathbf{x}' \in \mathcal{X} s.t. \hat{\mu}(\mathbf{x}') \leq \hat{\mu}(\mathbf{x}), \hat{\sigma}(\mathbf{x}') \leq \hat{\sigma}(\mathbf{x})\}$*. The hypervolume (HV) of the Pareto set* $\mathcal{P}'$ *with respect to a reference point* $\mathbf{r}$ *is* $HV(\mathcal{P}', \mathbf{r}) = \text{Volume}(\bigcup_{\mathbf{x}\in\mathcal{P}'}[\mathbf{x}, \mathbf{r}])$*. The contribution of a point* $\mathbf{x} \in \mathcal{S}$ *to the hypervolume of exploration-exploitation is then defined as* $\nabla HV(\mathbf{x}, \mathcal{P}', \mathbf{r}) = HV(\mathcal{P}', \mathbf{r}) - HV(\mathcal{P}' \backslash \{\mathbf{x}\}, \mathbf{r})$.

This metric evaluates the change in hypervolume resulting from the inclusion or exclusion of a sample point, thus measuring its contribution to both exploration and exploitation. The use of hypervolume as a measure is a well-established method in multi-objective optimization to evaluate the quality of solutions. It provides a clear and interpretable way to assess the impact of adding or removing points from the Pareto set. The exploration metric $\sigma(\mathbf{x})$ can be interpreted in different ways depending on the SO approach. When using a surrogate model, $\sigma(\mathbf{x})$ often represents the surrogate variance, which captures the model's uncertainty about the prediction at point $\mathbf{x}$. Alternatively, in a model-agnostic SO approach, $\sigma(\mathbf{x})$ can be represented by a distance metric $d(\mathbf{x})$, which measures the dissimilarity or novelty of the point $\mathbf{x}$ relative to other points in the dataset. This flexibility allows the framework to be adapted to various optimization scenarios and objectives.

3.2. Batch Properties Metrics

This category of metrics is dedicated to providing explanations for the selected batch of sample points. These metrics evaluate the batch based on its aggregated contribution to diversity (both intra-batch and inter-batch diversity), aggregated objective values, and its impact on the SO objective, specifically the exploration-exploitation trade-off.

Density of the selected subset (DES) is a metric that estimates the entropy of the batch. The entropy of a batch of sample points provides a measure of the uncertainty or randomness within the batch. In the context of surrogate optimization, higher entropy indicates greater diversity among the sample points, while lower entropy suggests that the points are more similar to each other. Entropy, in this context, can be calculated using techniques like Kernel Density Estimation (KDE)(Y.-C. Chen 2017), to estimate the underlying probability density function of a dataset without assuming a specific parametric form, making it suitable for complex distributions.

Definition 4. *For a batch of points $\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$, the DES metric can be defined using differential entropy. Let $\hat{p}(\mathbf{x})$ represent the KDE estimate of the probability density function for the points in $\mathcal{S}$. The differential entropy $H(\mathcal{S})$ is given by $H(\mathcal{S}) = -\int_{\mathbb{R}^d} \hat{p}(\mathbf{x}) \log \hat{p}(\mathbf{x})\, d\mathbf{x}$. In practice, for a finite sample, the differential entropy can be approximated as $H(\mathcal{S}) \approx -\frac{1}{k}\sum_{i=1}^{k} \log \hat{p}(\mathbf{x}_i)$, where $\hat{p}(\mathbf{x}_i)$ is the KDE estimate at point $\mathbf{x}_i$.*

By calculating the differential entropy of the batch, the DES metric provides insights into how well the batch balances exploration and exploitation. High DES values indicate a well-distributed set of sample points, contributing to effective exploration of the solution space.

Diversity of the selected subset (DIS) quantifies the spread or coverage of the selected set of points in a multidimensional space by measuring the volume of the geometric shape spanned the points. Inspired by the diverse sampling concept from Determinantal Point Processes (DPPs) (Kulesza, Taskar, et al. 2012), we aim to utilize the determinant of the kernel matrix L to measure the batch diversity.

Definition 5. *Given a batch of points $\mathcal{S}$, the diversity can be measured using the determinant of the kernel matrix L, where L_{ij} is computed using a similarity kernel function. $DIS(\mathcal{S})$ is then given by $DIS(\mathcal{S}) = \sum_{\sigma \in S_n} sign(\sigma) \prod_{i=1}^{n} L_{i,\sigma(i)}$, where σ is a permutation of the indices $1, 2, \dots, n$, and $\text{sign}(\sigma)$ is the sign of the permutation. When the kernel is a Radial Basis Function (RBF), $L_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$, (Steinwart, Hush, and Scovel 2006).*

Average Batch Distance (ABD): metric quantifies the dissimilarity between a batch of selected sample points and a set of previously evaluated data points. This metric is crucial for assessing the balance between exploration (distant points) and exploitation (close points).

Definition 6. *Let $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ be a set of n previously evaluated sample points, and let $\mathcal{S} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\}$ be a batch of k newly selected sample points. The ABD is defined as the average minimum Euclidean distance between each point in the selected batch $\mathcal{S}$ and the set of evaluated points $\mathcal{D}$: $ABD(\mathcal{S}, \mathcal{D}) = \frac{1}{k}\sum_{i=1}^{k} \min_{j=1}^{n} \| \mathbf{z}_i - \mathbf{x}_j \|$, where $\| \mathbf{z}_i - \mathbf{x}_j \|$ denotes the Euclidean distance between the points $\mathbf{z}_i$ and $\mathbf{x}_j$.*

Hypervolume of Exploration-Exploitation (HVE): metric measures the volume of the region dominated by the sample points in the selected batch. Hypervolume is a widely used metric in multi-objective optimization (Emmerich, Giannakoglou, and Naujoks 2006) to evaluate the quality of a set of points based on multiple objectives. We introduce this metric to explain the contribution of each batch to the exploration-exploitation trade-off in sampling for single-objective surrogate optimization.

Definition 7. *Consider a surrogate optimization approach characterized by an exploitation metric $\mu(\mathbf{x})$ and an exploration metric $\sigma(\mathbf{x})$. Let $r = (r_\mu, r_\sigma^2)$ be the reference point in the 2D objective space, where r_μ represents the reference value for exploitation, set to a sufficiently high value worse than any expected $\mu(\mathbf{x})$ r_σ^2 represents the reference value for exploration, set to a sufficiently high value worse than any expected $\sigma^2(\mathbf{x})$. Let $\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$ be a batch of k points selected in a given iteration. For each point $\mathbf{x}_i \in \mathcal{S}$, $\mu(\mathbf{x}_i)$ is the exploitation metric value at point $\mathbf{x}_i$ and $\sigma^2(\mathbf{x}_i)$ is the exploration metric value at point $\mathbf{x}_i$. The hypervolume contribution $HV(\mathbf{x}_i)$ of each point $\mathbf{x}_i$ is defined as the volume of the rectangular region dominated by $\mathbf{x}_i$ and bounded by the reference point r, $HV(\mathbf{x}_i) = \left(r_\mu - \mu(\mathbf{x}_i)\right) \cdot \left(r_\sigma^2 - \sigma^2(\mathbf{x}_i)\right)$. The total hypervolume contribution HV_S of the batch of k points is the sum of the individual hypervolume contributions $HV_S = \sum_{i=1}^{k} HV(\mathbf{x}_i) = \sum_{i=1}^{k} \left(r_\mu - \mu(\mathbf{x}_i)\right) \cdot \left(r_\sigma^2 - \sigma^2(\mathbf{x}_i)\right)$.*

Similar to Definition 3, the exploration metric $\sigma(\mathbf{x})$ can be interpreted in different ways depending on the surrogate optimization (SO) approach. When using a surrogate model, $\sigma(\mathbf{x})$ often represents the surrogate variance, capturing

the model's uncertainty about the prediction at point $\mathbf{x}$. Alternatively, in a model-agnostic SO approach, $\sigma(\mathbf{x})$ can be represented by a distance metric $d(\mathbf{x})$, measuring the dissimilarity or novelty of the point $\mathbf{x}$ relative to other points in the dataset. This metric provides a measure of the combined contribution of the selected batch to both exploration and exploitation objectives, enabling the evaluation of the effectiveness of the sampling strategy in balancing these two aspects in surrogate optimization.

3.3. Optimization Process Metrics

These metrics provide a holistic view of the overall optimization process, including convergence and performance.

Partitioning-Based Solution Space Analysis (PSSA) metric focuses on providing an explanation of the solution space, which is commonly considered as a d-dimensional hyperrectangular space. Accurately locating each sample point or the selected batch of points within this space is crucial prior to expensive evaluations. To achieve this, we utilize partitioning methods to divide the solution space into distinct regions, facilitating a better understanding of the distribution and relationships within the data.

Definition 8. *Let $\mathcal{F} = \{(\mathbf{x}_i, y_i) \mid i = 1,2,\dots,n\}$ be a d-dimensional dataset, where $\mathbf{x}_i \in \mathbb{R}^d$ represents an evaluated sample point and y_i is the corresponding expensive target value. A partitioning method divides the dataset into m regions, $\{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_m\}$, where each region $\mathcal{R}_k$ is a subset of the solution space. The region $\mathcal{R}_k$ can be defined as $\mathcal{R}_k = \{\mathbf{x} \in \mathcal{X} \mid \text{Condition}_k(\mathbf{x})\}$, where $\text{Condition}_k(\mathbf{x})$ specifies the criteria that determine the membership of $\mathbf{x}$ in region $\mathcal{R}_k$.*

As an example, consider decision trees as a partitioning method. In decision trees, the dataset is partitioned based on decision rules at each node, where each decision rule splits the data into two branches based on a threshold value of a specific feature. This process is repeated recursively, creating a hierarchical partitioning of the solution space. Each node in the decision tree splits the data based on a decision rule of the form $\mathbf{x}_j \leq v$, where $\mathbf{x}_j$ is the j-th feature and v is a threshold value. Each partition (leaf node) corresponds to a region $\mathcal{R}_k$ in the d-dimensional space. The region $\mathcal{R}_m$ associated with leaf node m can be defined as $\mathcal{R}_m = \{\mathbf{x} \in \mathcal{X} \mid \wedge_{(j,v)\in\mathcal{L}_m}(\mathbf{x}_j \leq v \text{ or } \mathbf{x}_j > v)\}$, where $\mathcal{L}_m$ is the set of splitting rules leading to leaf m.

By leveraging the explainability of decision trees, the PSSA metric systematically partitions the solution space and derives interpretable rules that elucidate the relationships between the features and the target values. This approach enables a clearer understanding of the solution space and facilitates the identification of regions that warrant further exploration or exploitation.

Convergence Rate (CR) tracks the rate at which the optimization process approaches the optimal solution. It is measured by the decrease in the objective function value over iterations.

Definition 9. *Given an objective function $f(\mathbf{x})$, let $\mathcal{I} = \{f(\mathbf{x}^{(1)}), f(\mathbf{x}^{(2)}), \dots, f(\mathbf{x}^{(T)})\}$ be the sequence of best known objective function values over T iterations. The Convergence Rate is defined as the average rate of decrease in the best known objective function value per iteration $CR = \frac{1}{T-1}\sum_{t=2}^{T}\left(\frac{f(\mathbf{x}^{(t-1)}) - f(\mathbf{x}^{(t)})}{f(\mathbf{x}^{(t-1)})}\right)$.*

This metric provides insight into how quickly the SO converges toward the optimal solution.

Optimization Stability (OS) metric evaluates the stability of the optimization process, ensuring it is not overly sensitive to initial conditions or random variations.

Definition 10. *Let $\mathcal{S}$ be a set of m optimization runs with different initial conditions or random seeds, resulting in objective function values $\{f(\mathbf{x}_1^*), f(\mathbf{x}_2^*), \dots, f(\mathbf{x}_n^*)\}$ at convergence. The Optimization Stability is defined as the standard deviation of the final objective function values at convergence $OS = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(f(\mathbf{x}_i^*) - \overline{f}\right)^2}$, where $\overline{f}$ is the mean of the objective function values at convergence $\overline{f} = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i^*)$.*

This metric quantifies the variability in the optimization outcomes, indicating the robustness.

3.4. Feature Importance

This group of metrics focuses on evaluating the significance of individual features in the context of surrogate optimization. By quantifying the contribution of each feature to various objectives and models, these metrics provide insights into the behavior and performance of the optimization process, enhancing its interpretability.

Feature Importance for Exploration and Exploitation (FIEE) is a metric that evaluates the contribution of each feature to the exploration and exploitation objectives.

Definition 11. *Given a set of input features* $\mathbf{x} = (x_1, x_2, \dots, x_d)$*, the contribution of each feature* x_j*,* $\forall j \in \{1, \dots, d\}$*, to the exploration objective is denoted by* $|\eta_j|$*, and to the exploitation objective by* $|\lambda_j|$*. These contributions can be calculated using feature importance techniques such as SHAP values, permutation importance, or other model-specific methods.* $\eta_j = \text{SHAP}_{\sigma}(x_j)$ or $\eta_j = \text{Permutation Importance}_{\sigma}(x_j), \lambda_j = \text{SHAP}_{\mu}(x_j)$ or $\lambda_j = \text{Permutation Importance}_{\mu}(x_j)$,
where $\text{SHAP}_{\sigma}(x_j)$ *and* $\text{Permutation Importance}_{\sigma}(x_j)$ *represent the SHAP value and permutation importance of feature* x_j *with respect to the exploration metric* $\sigma(\mathbf{x})$*, respectively, and* $\text{SHAP}_{\mu}(x_j)$ *and* $\text{Permutation Importance}_{\mu}(x_j)$ *represent the SHAP value and permutation importance of feature* x_j *with respect to the exploitation metric* $\mu(\mathbf{x})$*, respectively.*

Feature Importance for the Black-Box Objective Function (FIBB) is a metric that evaluates the contribution of each feature to the black-box objective function.

Definition 12. *The contribution of each feature* $j \in \{1, \dots, d\}$ *to the black-box objective function is denoted by* $|\phi_j|$*. This contribution can be calculated using data-driven feature importance methods such as SHAP values, permutation importance, or other relevant techniques,* $\phi_j = \text{SHAP}_f(x_j)$ or $\phi_j = \text{Permutation Importance}_f(x_j)$,
where $\text{SHAP}_f(x_j)$ *and* $\text{Permutation Importance}_f(x_j)$ *represent the SHAP value and permutation importance of feature* x_j *with respect to the black-box objective function* $f(\mathbf{x})$*.*
These metrics help in understanding the relative importance of features and their evolution across iterations, enhancing interpretability.

Feature Importance of Surrogate Model (FIS) is a metric that quantifies the contribution of individual features to the predictions generated by the surrogate model. It utilizes point-wise feature importance techniques, such as Shapley Values (Lundberg and Lee 2017), to offer a detailed analysis of how each feature influences the predicted outcomes.
Definition 13. *Given a surrogate model* $\hat{f}$ *and a set of input features* $\mathbf{x} = (x_1, x_2, \dots, x_d)$*, the contribution of each feature* j *to the surrogate model's prediction for a data point* $\mathbf{x}_i$ *can be quantified using feature importance scores* $\phi_j(\mathbf{x}_i)$*. The* ***FIS*** *metric for each feature* j *is then defined as the average feature importance score over all data points* $FIS_j = \frac{1}{n}\sum_{i=1}^{n} \phi_j(\mathbf{x}_i)$*, where* $\phi_j(\mathbf{x}_i)$ *is the feature importance score of feature* j *for the* i*-th data point, and* n *is the total number of data points. Feature importance scores* $\phi_j(\mathbf{x}_i)$ *can be obtained using various techniques, such as SHAP values, permutation importance, or other model-specific importance measures like Multivariate Adaptive Regression Splines (MARS) (Friedman 1991) and Decision Trees (DT) (Song and Ying 2015).*
We emphasize the advantages of using explainable surrogate models such as MARS or DT compared to other models like Gaussian Processes (GPs) (Rasmussen and Nickisch 2010). Although GPs are powerful for capturing complex, non-linear relationships, they lack inherent mechanisms for directly extracting feature importance.

4. Results

In this section, we elaborate on the impact and applicability of the proposed explanation metrics, employing the well-known synthetic and real-world benchmark problems from black-box optimization literature.

Baselines: We consider the standard Monte-Carlo-based batch Bayesian acquisition functions such as **qEI**, **qUCB**, **qMES**, and **qGibbon** as described in (Wilson et al. 2017; Balandat et al. 2020). Moreover, we consider **DEEPA** (Nezami and Anahideh 2024) as a Pareto-based SO baseline which has been shown to be effective for high-dimensional BBO problems.

Experiment Set-up: In our experiments, the size of the initial evaluated set is assigned based on the $2 * (d + 1)$ formula, where d refers to the number of features. For the $10d$ test problems, we maintain a batch size of $k = 8$, a candidate set of 1000 randomly generated sample points. For the $2d$ instances, we use a batch size of $k = 4$, a candidate set of 100 randomly generated sample points. For the Levy $6d$ test problem, the batch size is set to 4, and the candidate set includes 1000 points. For the Robot Pushing and Rover Trajectory benchmarks, the batch size is set to $k = 50$.

Figures 1 and 2 demonstrate how **Batch Properties Metrics** provide valuable insights, enabling the user to compare batch selections across different baselines. For the 10-dimensional Rastrigin test problem, the ABD plot indicates that **qGibbon** tends to select points that are more distant from the evaluated set for batch evaluation compared to other baselines. This characteristic makes **qGibbon** an excellent choice for users prioritizing exploration. This behavior aligns with the Gibbon acquisition function, which is equivalent to a Determinantal Point Process (DPP) when specific quality functions and similarity kernels are used. Maximizing the determinant of the selected subset ensures that a diverse set of points is chosen for evaluation.

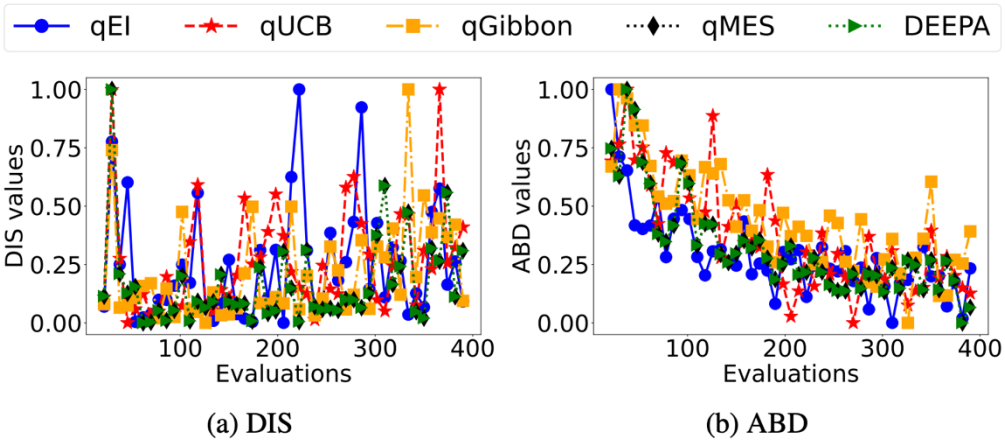


Figure 1. Batch Properties—Rastrigin 10D

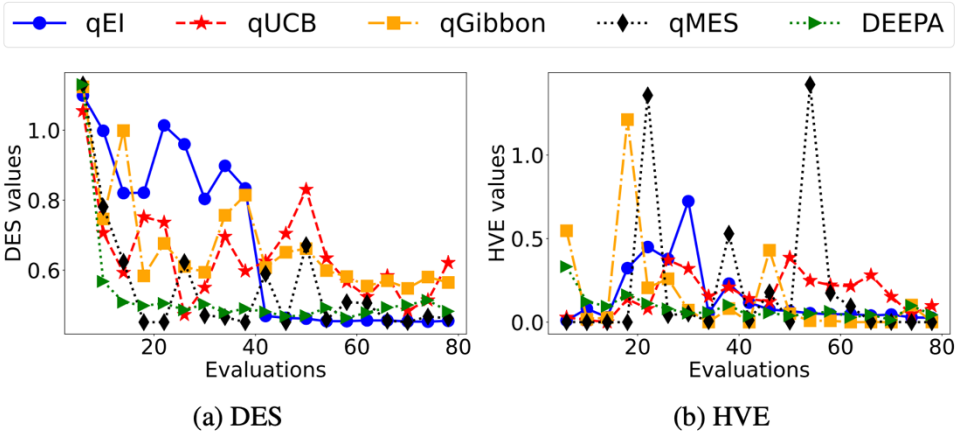


Figure 2. Batch Properties—Rosenbrock 2D

Figure 3 presents the PCE and MDPE from the **Sampling Core Metrics** for the 14-dimensional Robot Pushing problem. By leveraging PCE plots during local convergence, users can actively guide the optimization process to select points with more diverse coordinate values, thereby reducing the risk of budget inefficiency. For instance, the **DEEPA** strategy tends to select sample points with a higher range of values in the second dimension compared to the 14th dimension in the final iterations. Additionally, the MDPE plot proves valuable at any stage of the optimization

process, providing a clear visualization of the point-wise distances between the evaluated set and the sample points chosen by a given acquisition function. For example, in the last few iterations, **qMES** tends to focus on points that are closer (with lower distance values) compared to other baselines, making it more exploitative.

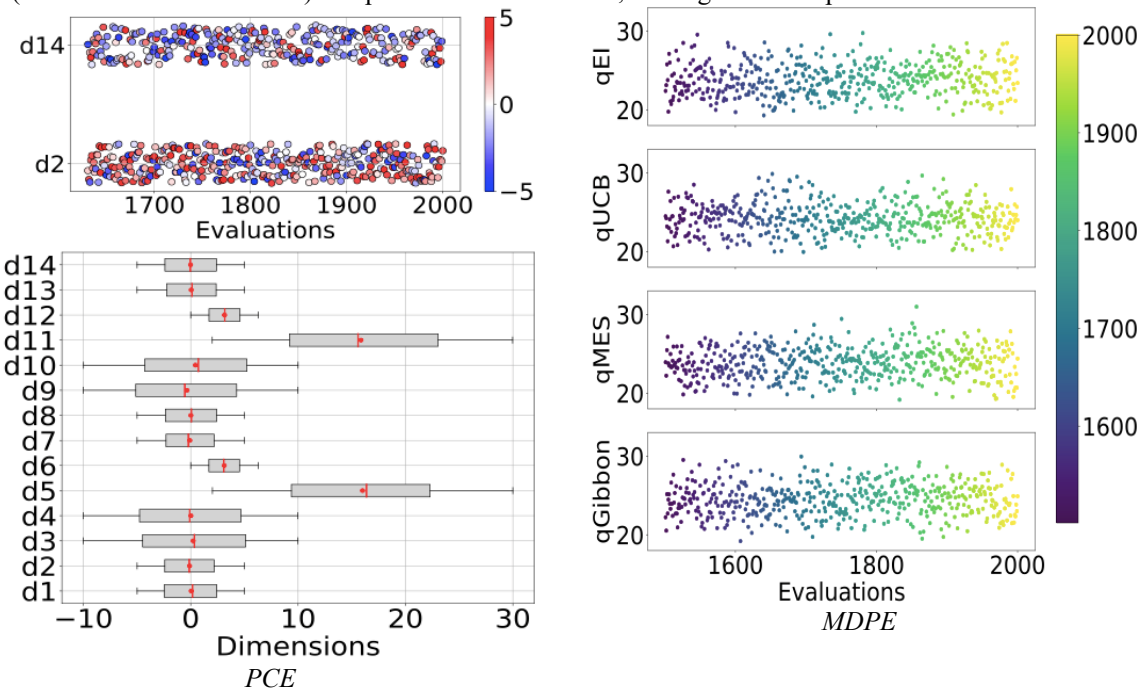


Figure 3. Sampling Core—Robot Pushing 14D

```
|--- d12 <= 5.68
|   |--- d8 <= 4.12
|   |   |--- d4 <= 8.52
|   |   |   |--- value: [-0.01]
|   |   |--- d4 >  8.52
|   |   |   |--- value: [1.19]
|   |--- d8 >  4.12
|   |   |--- d13 <= -3.45
|   |   |   |--- value: [-0.81]
|   |   |--- d13 >  -3.45
|   |   |   |--- value: [-0.99]
|--- d12 >  5.68
|   |--- d13 <= 1.10
|   |   |--- d1 <= -2.39
|   |   |   |--- value: [-1.27]
|   |   |--- d1 >  -2.39
|   |   |   |--- value: [-2.52]
|   |--- d13 >  1.10
|   |   |--- value: [0.07]
```

(a) DT rules — 1st Iteration

```
|--- d4 <= -4.91
|   |--- d3 <= 9.68
|   |   |--- d10 <= 9.83
|   |   |   |--- value: [-0.52]
|   |   |--- d10 >  9.83
|   |   |   |--- value: [-7.98]
|   |--- d3 >  9.68
|   |   |--- d12 <= 2.01
|   |   |   |--- value: [-0.63]
|   |   |--- d12 >  2.01
|   |   |   |--- value: [-20.19]
|--- d4 >  -4.91
|   |--- d11 <= 29.06
|   |   |--- d10 <= -8.06
|   |   |   |--- value: [-0.65]
|   |   |--- d10 >  -8.06
|   |   |   |--- value: [-0.01]
|   |--- d11 >  29.06
|   |   |--- d3 <= 8.11
|   |   |   |--- value: [-0.47]
|   |   |--- d3 >  8.11
|   |   |   |--- value: [-10.13]
```

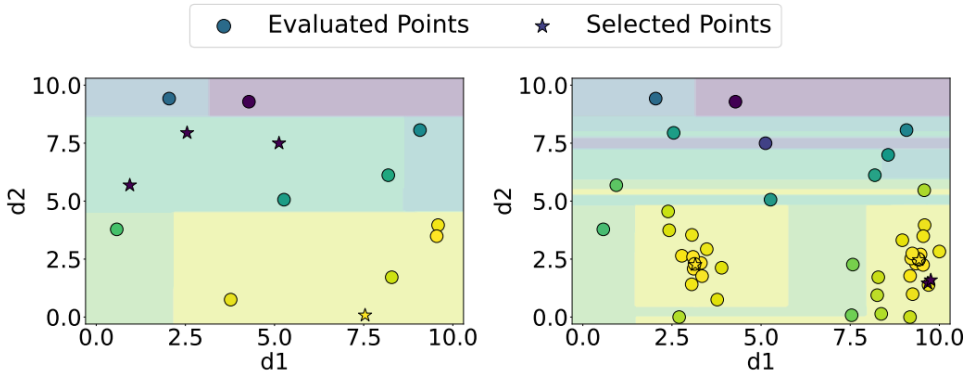
(b) DT rules — 30th Iteration

Figure 4. PSSA—Robot Pushing 14D

Figures 4 and 5 depict the PSSA from the **Optimization Process Metrics** for the Branin and Robot Pushing problems. PSSA enables users to monitor promising regions within the solution space (highlighted in yellow in Figure 5) and locate newly selected sample points by revealing the partitioning rules (Figure 4). For example, in the Robot Pushing problems, we can calculate the number of points selected from each partition based on the Decision Tree (DT) rules in each iteration ($k = 50$). Additionally, PSSA assists users in adjusting the sampling strategy if excessive exploitation or exploration is detected within specific areas of the solution space. Moreover, plots in Figure 6 provide insight into

the performance of different SO baselines. For the 6-dimensional Levy, **DEEPA** outperforms other baselines in terms of convergence rate, while **qGibbon** shows greater stability (less variation) across the final iterations.

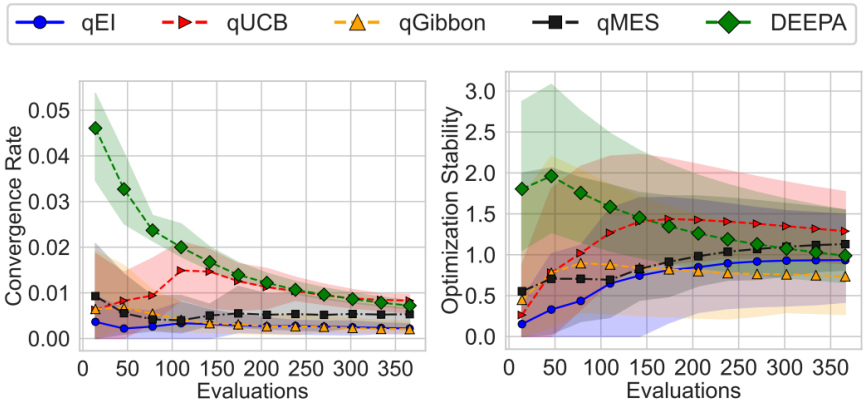
Figures 7a and 7b show the top 10 features (coordinates) utilizing FIBB and FIEE (exploration) from the **Feature Importance Metrics** for the high-dimensional Rover Trajectory test problem. For example, dimensions 32 and 19 significantly contribute to the BBO exploration objective (Maxmin distance) while the first and last 3 dimensions have more impact on the true black-box objective value.



(a) Partitions — 1st iteration (b) Partitions — 10th iteration

Figure 5. PSSA—Branin 2D

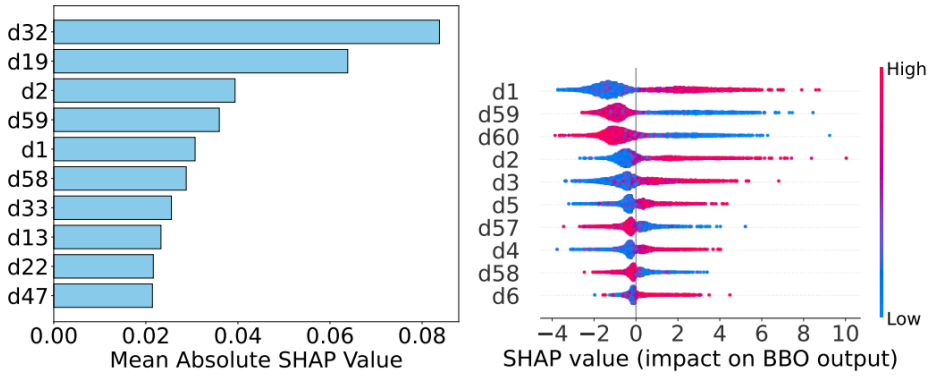
Moreover, Figures 8a and 8b show the FIS metric for the 6-dimensional Levy test problem. Figure 8a utilizes GP as the surrogate model and calculates MARS feature importance based on the observed data in each iteration. Figure 8b uses MARS, which is an explainable surrogate model, and utilizes the surrogate's feature importance in each iteration. The user can utilize these results to monitor how the surrogate's important coordinates change across iterations.



(a) CR (b) OS

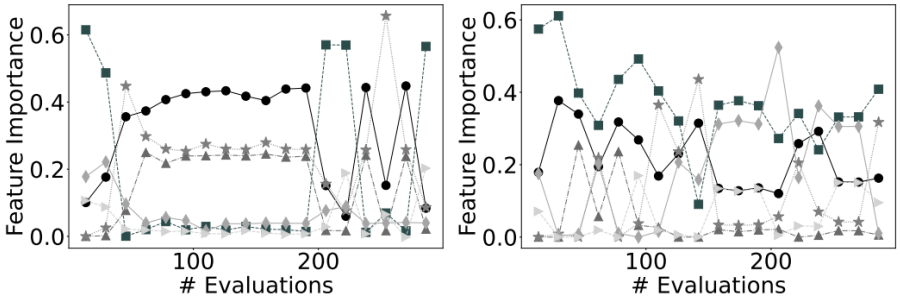
Figure 6. Optimization Process—Levy 6D

Practical Applications of IEMSO Metrics:
To further illustrate the application of IEMSO metrics, we examine two comparative examples of using **qEI** and **qUCB** on the same initialization run of the real-world Robot Pushing problem and Levy test problem. Figures 9 and 10 present the batch property metrics, including DIS, ABS, and HVE, alongside the typical convergence plot, which reflects the best-observed solution after each iteration.



(a) FIEE Top 10 features (b) FIBB Top 10 features

Figure 7. Feature Importance—Rover 60D (DEEPA)



(a) Srg: GP — FI: MARS (b) Srg: MARS — FI: MARS

Figure 8. FIS—Levy 6D (DEEPA)

In standard practice, SO practitioners primarily rely on convergence pattern, as shown in Figure 9a and Figure 10a, which depicts the best-known solution and the corresponding improvement in the black-box objective following expensive evaluations. By using these performance-based plots—alternatively, regret could be plotted—users can compare the outcomes of **qEI** and **qUCB**. In both examples, it is evident that **qUCB** achieves faster convergence and a superior final solution, making it more suitable for these applications in the Robot Pushing and Levy test problem. However, the reasoning behind this performance difference and the sampling logic driving the gap often remains unclear.

Using batch property metrics derived from IEMSO, such as HVE, ABD, and DIS, as shown in Figures 9b to 9d, users can gain deeper insights into the results obtained in the Robot Pushing problem. For instance, the HVE metric indicates that **qUCB** better maintains the exploration-exploitation trade-off compared to **qEI**. Similarly, DIS and ABD metrics reveal that **qUCB** emphasizes exploration more as iterations progress, while the exploratory behavior of Expected Improvement diminishes over time. Additionally, **qUCB** exhibits higher exploitation early on, whereas **qEI** prioritizes exploration due to greater uncertainty values. Consequently, **qUCB** selects points that are farther from the evaluated set and are more diverse than those chosen by **qEI**.

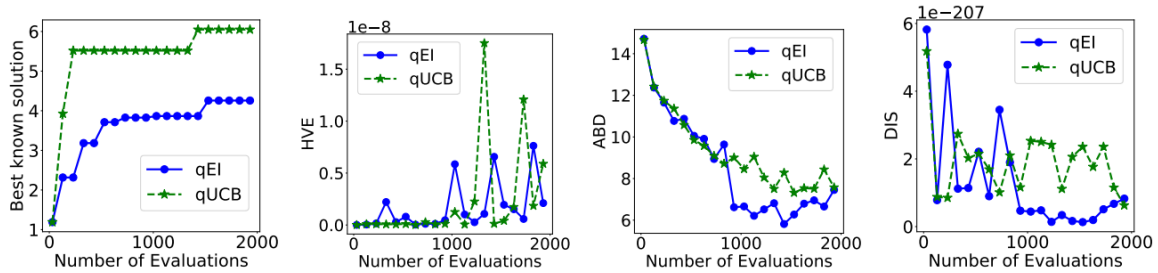


Figure 9. Batch Properties Metrics for Performance Comparison: Robot Pushing Problem (14D)

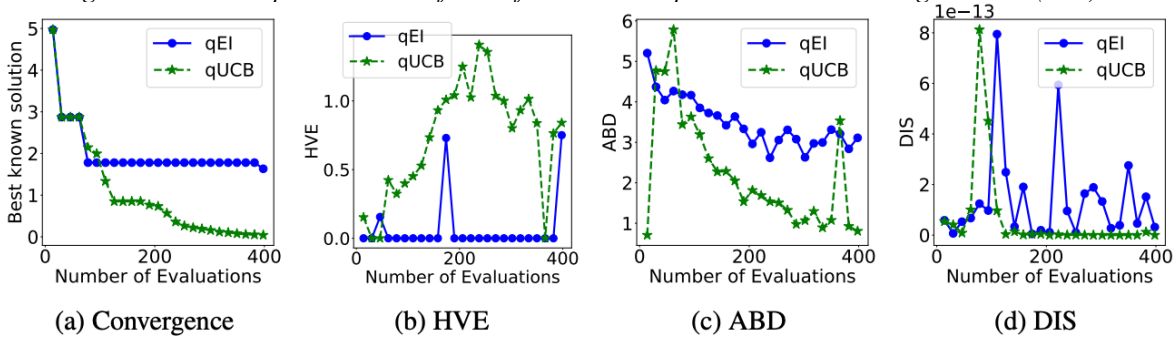


Figure 10. Batch Properties Metrics for Performance Comparison: Levy Test Problem (6D)

In the context of the Robot Pushing problem, the enhanced batch diversity and exploration by **qUCB** significantly improve the exploration-exploitation balance, leading to its superior performance over **qEI**. However, this logic does not extend to interpreting the results obtained for the Levy problem. In contrast, the IEMSO batch property metrics such as HVE, ABD, and DIS, as shown in Figures 10b to Figure [levy-dis], reveal a different pattern. While **qUCB** maintains a better exploration-exploitation trade-off, **qEI** demonstrates increased exploration compared to **qUCB** as the iterations progress. Instead of focusing on promising regions, **qEI** prioritizes reducing uncertainty and exploring more extensively. This behavior ultimately leads to **qUCB** outperforming **qEI** in this scenario as well.

5. Conclusion

In this paper, we introduced Inclusive Explainability Metrics for Surrogate Optimization (IEMSO), a comprehensive framework designed to enhance transparency and build trust in black-box optimization (BBO) methods. Our approach addresses significant gaps in the current state of explainability for surrogate optimization, providing a set of model-agnostic metrics that can be applied across different surrogate frameworks and optimization scenarios. Through extensive experimental analysis on both synthetic and real-world benchmark problems, we demonstrated the effectiveness of IEMSO in offering valuable insights into various stages of the optimization process. The metrics were divided into four main categories: Sampling Core Metrics, Batch Properties Metrics, Optimization Process Metrics, and Feature Importance Metrics. Each category serves a unique purpose in elucidating different aspects of surrogate optimization, from explaining sampling strategies to understanding feature importance and evaluating the overall optimization process. Our results show that IEMSO not only enhances the explainability of surrogate optimization outcomes but also enables experts to understand the reason behind the performance difference among various approaches. By offering both intermediate and post-hoc explanations, IEMSO ensures that the decision-making processes are transparent, interpretable, and trustworthy, thereby increasing the reliability and applicability of surrogate optimization in complex, real-world scenarios. Future work could focus on expanding the set of explainability metrics and developing a user-friendly interface to further enhance the tool's applicability.

References

Adachi, M., Planden, B., Howey, D. A., Maundet, K., Osborne, M. A., & Chau, S. L. Looping in the Human: Collaborative and Explainable Bayesian Optimization. *arXiv Preprint arXiv:2310.17273*, 2023.

Angermueller, C., Belanger, D., Gane, A., Mariet, Z., Dohan, D., Murphy, K., Colwell, L., & Sculley, D. Population-Based Black-Box Optimization for Biological Sequence Design. In *International Conference on Machine Learning*, 324–334. PMLR, 2020.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., et al. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges Toward Responsible AI. *Information Fusion*, 58, 82–115, 2020.

Balandat, M., Karrer, B., Jiang, D., Daulton, S., Letham, B., Wilson, A. G., & Bakshy, E. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. *Advances in Neural Information Processing Systems*, 33, 21524–21538, 2020.

Carvalho, D. V., Pereira, E. M., & Cardoso, J. S. Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics*, 8(8), 832, 2019.

Chakraborty, T., Seifert, C., & Wirth, C. Explainable Bayesian Optimization. *arXiv Preprint arXiv:2401.13334*, 2024.

Chen, W., Chiu, K., & Fuge, M. Aerodynamic Design Optimization and Shape Exploration Using Generative Adversarial Networks. In *AIAA Scitech 2019 Forum*, 2351, 2019. https://doi.org/10.2514/6.2019-2351

Chen, Y. C. A Tutorial on Kernel Density Estimation and Recent Advances. *Biostatistics & Epidemiology*, 1(1), 161–187, 2017.

Desautels, T., Krause, A., & Burdick, J. W. Parallelizing Exploration-Exploitation Tradeoffs in Gaussian Process Bandit Optimization. *Journal of Machine Learning Research*, 15, 3873–3923, 2014.

Emmerich, M. T. M., Giannakoglou, K. C., & Naujoks, B. Single- and Multiobjective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4), 421–439, 2006.

Frazier, P. I. A Tutorial on Bayesian Optimization. *arXiv Preprint arXiv:1807.02811*, 2018.

Friedman, J. H. Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19(1), 1–67, 1991.

Jones, D. R., Schonlau, M., & Welch, W. J. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4), 455–492, 1998.

Kulesza, A., & Taskar, B., et al. Determinantal Point Processes for Machine Learning. *Foundations and Trends in Machine Learning*, 5(2–3), 123–286, 2012.

Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy*, 23(1), 18, 2020.

Lundberg, S. M., & Lee, S. I. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, 30, 2017.

Nezami, N., & Anahideh, H. Dynamic Exploration–Exploitation Pareto Approach for High-Dimensional Expensive Black-Box Optimization. *Computers & Operations Research*, 166, 106619, 2024.

Rasmussen, C. E., & Nickisch, H. Gaussian Processes for Machine Learning (GPML) Toolbox. *The Journal of Machine Learning Research*, 11, 3011–3015, 2010.

Ribeiro, M. T., Singh, S., & Guestrin, C. "Why Should I Trust You?" Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144, 2016.

Rodemann, J., Croppi, F., Arens, P., Sale, Y., Herbinger, J., Bischl, B., Hüllermeier, E., Augustin, T., Walsh, C. R., & Casalicchio, G. Explaining Bayesian Optimization by Shapley Values Facilitates Human-AI Collaboration. *arXiv Preprint arXiv:2403.04629*, 2024.

Seitz, S. Gradient-Based Explanations for Gaussian Process Regression and Classification Models. *arXiv Preprint arXiv:2205.12797*, 2022.

Sharma, N. A., Chand, R. R., Buksh, Z., Ali, A. B. M. S., Hanif, A., & Beheshti, A. Explainable AI Frameworks: Navigating the Present Challenges and Unveiling Innovative Applications. *Algorithms*, 17(6), 227, 2024.

Singh, M., Brownlee, A. E. I., & Cairns, D. Towards Explainable Metaheuristic: Mining Surrogate Fitness Models for Importance of Variables. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 1785–1793, 2022.

Song, Y. Y., & Ying, L. Decision Tree Methods: Applications for Classification and Prediction. *Shanghai Archives of Psychiatry*, 27(2), 130, 2015.

Srinivas, N., Krause, A., Kakade, S. M., & Seeger, M. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. *arXiv Preprint arXiv:0912.3995*, 2009.

Steinwart, I., Hush, D., & Scovel, C. An Explicit Description of the Reproducing Kernel Hilbert Spaces of Gaussian RBF Kernels. *IEEE Transactions on Information Theory*, 52(10), 4635–4643, 2006.

Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., & Guyon, I. Bayesian Optimization Is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track*, 3–26. PMLR, 2021.

Wilson, J. T., Moriconi, R., Hutter, F., & Deisenroth, M. P. The Reparameterization Trick for Acquisition Functions. *arXiv Preprint arXiv:1712.00424*, 2017.

Biographies

Nazanin Nezami is a researcher at the University of Illinois Chicago. Her research interests include black-box optimization, surrogate modeling, and explainability in machine learning-driven decision systems.

Hadis Anahideh is an Assistant Professor at the University of Illinois Chicago. Her research focuses on black-box optimization, sequential decision-making under uncertainty, responsible machine learning, and explainable AI.