Exploring Exploration in Bayesian Optimization

Leonard Papenmeier[*1] **Nuojin Cheng**[*2] **Stephen Becker**[2] **Luigi Nardi**[1, 3]

[1]Department of Computer Science, Lund University, Lund, Sweden
[2]Department of Applied Mathematics, University of Colorado Boulder, Boulder, Colorado, USA
[3]DBtune

Abstract

A well-balanced exploration-exploitation trade-off is crucial for successful acquisition functions in Bayesian optimization. However, there is a lack of quantitative measures for exploration, making it difficult to analyze and compare different acquisition functions. This work introduces two novel approaches – observation traveling salesman distance and observation entropy – to quantify the exploration characteristics of acquisition functions based on their selected observations. Using these measures, we examine the explorative nature of several well-known acquisition functions across a diverse set of black-box problems, uncover links between exploration and empirical performance, and reveal new relationships among existing acquisition functions. Beyond enabling a deeper understanding of acquisition functions, these measures also provide a foundation for guiding their design in a more principled and systematic manner.

1 INTRODUCTION

Bayesian optimization (BO) is a widely used method to maximize black-box functions. Given a function $f : \mathcal{X} \to \mathbb{R}$, BO guides the optimization process by sequentially constructing probabilistic surrogates – typically a Gaussian process (GP) – and selecting new sampling points by maximizing an acquisition function (AF) $\alpha : \mathcal{X} \to \mathbb{R}$. The AF is a key component for any BO algorithm that chooses the point to evaluate next. As BO is typically used for problems where the underlying function is expected to be multi-modal, it is crucial that the AF exhibits *explorative* behavior, allowing it to discover different modes of the objective function, but also *exploitative* behavior, allowing it to focus on finding the optimum in a promising region of the search space $\mathcal{X}$.

[*]Equal contribution.

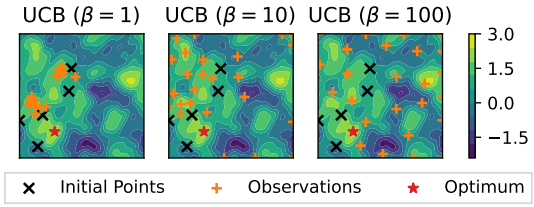


Figure 1: Observations of UCB with various β values (1, 10, and 100) on a two-dimensional GP-prior sample reveal the explorative behavior for different β. The black crosses are initial points, the orange plus signs are the observations of the BO phase, and the red star is the optimal location.

In short, a successful AF should exhibit a good exploration-exploitation trade-off (EETO) that balances these desiderata.

It is widely recognized in the BO community that different AFs exhibit varying degrees of explorative preference, and some AFs include parameters that allow explicit control over this behavior. For example, upper confidence bound (UCB) [Srinivas et al., 2010] employs a parameter β to regulate the level of exploration. Figure 1 illustrates an example of UCB with various β values applied to a two-dimensional GP prior with a length scale of 0.1. In this experiment, five initial points were fixed, and observations were collected over 25 iterations for each β value. A larger β produces a more dispersed layout of observations, indicating increased exploration. Similar behavior is observed in other acquisition functions, such as weighted expected improvement [Sóbester et al., 2005] and ϵ-greedy strategies [Sutton, 2018]. However, when comparing AFs from different families – such as UCB versus weighted expected improvement or ϵ-greedy strategies – assessing their exploration preferences becomes challenging, as no universally accepted metric exists to quantify these characteristics. Understanding the exploration tendencies of different AFs is important, as this knowledge influences the selection of appropriate AFs for real-world problems, especially when a specific level of exploration is required.

Proceedings of the 41st Conference on Uncertainty in Artificial Intelligence (UAI 2025), PMLR 244:3388–3415.

In this work, we fill this gap by proposing two novel quantities to quantify the level of exploration. We make the following contributions:

- We propose two novel means for quantifying exploration named observation traveling salesman distance (OTSD) and observation entropy (OE)[1]. The first quantity is based on the total Euclidean distance of a traveling salesman tour connecting all the observation points in the search space, while the second adopts an information-theoretic approach and uses the empirical differential entropy of the observations.
- We introduce the first empirical taxonomy of acquisition function exploration based on these new methods, demonstrating their effectiveness in capturing exploration behavior.
- We provide an extensive evaluation across a diverse set of low- and high-dimensional synthetic and real-world benchmarks, demonstrating the exploration behavior of popular acquisition functions and their extensions. OTSD and OE strongly correlate in benchmark problems, cross-validating their reliability.

2 BACKGROUND AND RELATED WORK

2.1 GAUSSIAN PROCESSES

A GP is a stochastic process that models an unknown function. It is characterized by the property that any finite set of function evaluations follows a multivariate Gaussian distribution. Assuming that the prior has a zero mean, a GP is uniquely determined by the current observations $\mathcal{D}_t \coloneqq \{(\boldsymbol{x}_i, y_{\boldsymbol{x}_i})\}_{i=1}^t$ and the kernel function $\kappa(\boldsymbol{x}, \boldsymbol{x}')$. Given these, at stage t, the predicted mean of $y_{\boldsymbol{x}}$ at a new point $\boldsymbol{x}$ is $\mu_t(\boldsymbol{x}) = \boldsymbol{\kappa}_t(\boldsymbol{x})^T(\boldsymbol{K}_t)^{-1}\boldsymbol{y}_t$, and the predicted covariance between points $\boldsymbol{x}$ and $\boldsymbol{x}'$ is $\mathrm{Cov}_t(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{\kappa}_t(\boldsymbol{x})^T(\boldsymbol{K}_t + \tilde{\sigma}\boldsymbol{I})^{-1}\boldsymbol{\kappa}_t(\boldsymbol{x}')$, where $[\boldsymbol{\kappa}_t(\boldsymbol{x})]_i = \kappa(\boldsymbol{x}_i, \boldsymbol{x})$, $[\boldsymbol{y}_t]_i = y_{\boldsymbol{x}_i}$, $\tilde{\sigma}$ is noise level, and $[\boldsymbol{K}_t]_{i,j} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$. At point $\boldsymbol{x}$, the GP posterior $y_{\boldsymbol{x}} \sim \mathcal{N}(\mu_t(\boldsymbol{x}), \sigma_t(\boldsymbol{x}))$, where $\sigma_t(\boldsymbol{x}) \coloneqq \mathrm{Cov}_t(\boldsymbol{x}, \boldsymbol{x})$; see Rasmussen et al. [2006] for more details.

2.2 ACQUISITION FUNCTIONS

One-step. One commonly used AF for BO is expected improvement (EI) [Jones et al., 1998], defined as $\alpha_{\mathrm{EI}}(\boldsymbol{x}) \coloneqq \mathbb{E}\big[\max\{y_{\boldsymbol{x}} - y_t^*, 0\}\big]$, where y_t^* denotes the current-best (i.e., incumbent) observation. A closely related function is probability of improvement (PI) [Jones, 2001], which considers only the probability that a new observation $y_{\boldsymbol{x}}$ exceeds the incumbent y_t^*, without accounting for the magnitude of improvement: $\alpha_{\mathrm{PI}}(\boldsymbol{x}) \coloneqq \mathbb{P}\big[y_{\boldsymbol{x}} > y_t^*\big]$. Both EI and PI have closed-form expressions and are therefore computationally efficient. Similarly, UCB is defined as $\alpha_{\mathrm{UCB}}(\boldsymbol{x}) \coloneqq \mu_t(\boldsymbol{x}) + \sqrt{\beta_t}\,\sigma_t(\boldsymbol{x})$, where β_t is a parameter that balances exploration and exploitation. In contrast, information-theoretic AFs select the next sampling point to reduce uncertainty about a particular property of the optimum – whether its location as in entropy search (ES) or predictive entropy search (PES) [Hennig and Schuler, 2012, Hernández-Lobato et al., 2014], its value as in max-value entropy search (MES) [Wang and Jegelka, 2017], or both as in joint entropy search (JES) [Hvarfner et al., 2022, Tu et al., 2022]. Thompson sampling (TS) implicitly balances the exploration-exploitation trade-off by maximizing posterior samples from a Gaussian process whose accuracy improves as more observations are incorporated [Bijl et al., 2016]. However, TS has been criticized for being overly explorative. To address this, several approaches have been proposed to make it less explorative, including trust regions (TRs) that restrict the space over which the AF is maximized to a subregion of $\mathcal{X}$ [Eriksson et al., 2019] and random axis-aligned subspace perturbations (RAASP) sampling that only considers points close to the incumbent observation for the maximization of the AF [Rashidi et al., 2024].

Multi-step. One common property of all aforementioned AFs is that they assume that the next evaluation will be the last, i.e., they greedily maximize the simple or inference regret for the next iteration, assuming no more evaluations will be performed [Wang and Jegelka, 2017]. In contrast, multi-step AFs [Ginsbourger and Le Riche, 2010, Wu and Frazier, 2019, Jiang et al., 2020] consider the impact of the current choice for future evaluations: $\alpha_{\mathrm{MS}}(\boldsymbol{x}) = v_1(\boldsymbol{x}|\mathcal{D}_t) + \mathbb{E}_y\left[\max_{\boldsymbol{x}_2}\left(v_1(\boldsymbol{x}_2|\mathcal{D}_t \cup \{(\boldsymbol{x}, y_{\boldsymbol{x}})\} + \mathbb{E}_{y_2}\left[\ldots\right]\right)\right]$, where $v_1(\boldsymbol{x})$ is the one-step marginal value of $\boldsymbol{x}$, e.g., the expected improvement upon observing $\boldsymbol{x}$. See Jiang et al. [2020] for details. Multi-step AFs, such as knowledge gradient (KG) [Frazier, 2009], are computationally expensive since the expectations of α_{MS} must be approximated with Monte-Carlo methods and, therefore, are often limited to one lookahead step even though the theoretical framework can usually be extended to arbitrarily many lookahead steps [Jiang et al., 2020]. At the same time, they are fundamentally different from previous AFs and may be characterized by a unique EETO [Wu and Frazier, 2019].

Batch. Batching is a technique used where multiple function evaluations can be performed in parallel. Instead of re-conditioning the GP after a single new observation, one observes q points in parallel. Batching requires modifications of the AF to ensure that a batch contains a diverse set of candidates. One strategy for batch BO is using multi-point AFs that estimate the improvement of some utility upon observing q new points [Wang et al., 2020]. Other approaches include *local penalization* [González et al., 2016] that repels points from regions around points already included in the

[1] Our code is available at `https://github.com/LeoIV/exploring-exploration-public`

batch.

2.3 EXPRESSIONS OF EXPLORATION

Quantifying Exploration. Although achieving a good balance between exploration and exploitation is widely recognized as a crucial component of an effective acquisition function, the BO community still lacks a simple and convenient metric for quantifying exploration. Several related metrics have been proposed – such as total center distance (TCD) [Eriksson et al., 2019, Fig. 6] and incumbent distance [Hvarfner et al., 2024, Fig. 20] – but each has notable shortcomings. For instance, TCD focuses solely on the movement of the incumbent, and it fails when the first sample lands on the optimum (yielding a TCD of zero) or when the search oscillates between multiple optima. Similarly, incumbent distance measures the distance between the next query and the incumbent but does not reliably capture exploration, as repeatedly querying the opposite corner of the space can yield high values without representing meaningful explorative behavior.

Another approach employs Pareto analysis to interpret the EETO [Bischl et al., 2014, Feng et al., 2015, Žilinskas and Calvin, 2019, De Ath et al., 2021]. In this framework, exploration and exploitation are treated as distinct objectives, with the utility function depending on both the predicted mean μ and variance σ. Under this interpretation, many AFs, such as PI, EI, and UCB, can be accommodated. However, more sophisticated AFs, particularly those based on information-theoretic principles, do not conform to this framework.

Quantifying Exploration in Evolutionary Algorithms. The measurement of the exploration extends beyond Bayesian optimization. In evolutionary algorithms (EAs) [Eiben and Smith, 2015], this quantity is evaluated from two perspectives: genotypic and phenotypic [Črepinšek et al., 2013]. Genotypic measures assess diversity in the input space using tree-based techniques [Burke et al., 2002, Črepinšek et al., 2011], Euclidean distance quantities [McGinley et al., 2011], entropy-based methods [Misevičius, 2011], and individual-population similarity indices [Inoue et al., 2015]. In contrast, phenotypic measures evaluate diversity in terms of fitness or performance, employing distance-based methods [Chaiyaratana et al., 2007], entropy-based measures [Adra and Fleming, 2010, Turkey and Poli, 2014], and local-attraction metrics [Jerebic et al., 2021]. Notably, the distance-based approaches mentioned above typically measure the distance from each individual to the population's center rather than the aggregate distance connecting all individuals, distinguishing them from our method. Additionally, entropy-based approaches have mainly focused on low-dimensional discrete domains – often using binning methods to approximate entropy – whereas our proposed techniques target high-dimensional continuous domains.

Quantifying Exploration in Reinforcement Learning. In reinforcement learning (RL), whether in simpler settings like multi-armed bandits (MABs), a special case of Markov decision processes (MDPs) where the state remains constant, or in more complex MDP scenarios, an agent must balance exploration and exploitation to achieve long-term benefits.

In MAB problems, one measure of exploration is tracking the frequency with which each action (arm) is selected [Kuleshov and Precup, 2014]. For general RL tasks, both states and actions must be considered. Some methods promote exploration by maximizing the entropy of the action distribution [Williams and Peng, 1991, Ahmed et al., 2019] to encourage the agent to try diverse actions. Curiosity-driven exploration, often quantified using information gain [Sun et al., 2011, Houthooft et al., 2016], provides another approach to exploration in the state-action space. Existing entropy-based exploration methods in RL often rely on parametric methods, like variational inference, which leverage prior knowledge of the underlying density functions (e.g., for actions) but may be inaccurate when the parametric assumptions fail. In BO, we do not have access to the true density of observations, so we adopt a non-parametric approach for density estimation. Consequently, our method is not directly comparable to previous entropy-based approaches in RL that depend on specific parametric families.

Low-Discrepancy Sequences. Low-discrepancy sequences, also known as quasi-random sequences, are designed to fill a space more uniformly than random sampling. They are often used in numerical integration and optimization tasks to ensure that samples are evenly distributed across the search space. Examples include Sobol sequences [Sobol, 1967] and Halton sequences [Halton, 1964]. Discrepancy measure themselves, such as the star discrepancy [Niederreiter, 1992], quantify how uniformly a set of points covers a space and thus can be used to assess exploration. However, current methods are expensive to compute and finding more efficient methods is an active area of research [Clément et al., 2023].

3 QUANTIFYING EXPLORATION

Motivated by the Cambridge dictionary definition of *exploration* as "the activity of travelling to and around a place, especially one where you have never been [...] before, in order to find out more about it", we define exploration in the context of black-box optimization.

Definition 3.1 (Exploration)**.** The activity of sampling in a region of the search space, especially one that has never been sampled before, to learn more about a global optimum.

Quantifying the exploration preferences of AFs is crucial for understanding their behavior and for developing an effective AF portfolio to achieve better performance. In this section, we first summarize existing knowledge on the exploration tendencies of different acquisition functions and then propose two key methods to quantify exploration.

3.1 ANALYSIS OF TRIBAL KNOWLEDGE

The EI and PI AFs have been shown to explore relatively little [De Ath et al., 2021], with PI being even less explorative than EI [Benjamins et al., 2022]. In contrast, the KG AF – which can be seen as a generalization of EI [Wu, 2017, p. 12] – has been reported to be more explorative than EI [Frazier, 2009, p. 89]. Information-theoretic acquisition functions are generally considered on the explorative side [Hernández-Lobato et al., 2014], although they can be surpassed by UCB with a high β value. At the extremes of the spectrum, random search (RS) samples points uniformly at random throughout the domain $\mathcal{X}$, while deterministic selection (DM) always selects the same fixed point; both completely disregard the probabilistic surrogate model. Empirical findings on the exploration behavior of AFs can be broadly summarized by the following informal ordering: RS $\succeq$ UCB (high β) $\succeq$ Information Theoretic $\overset{?}{\succeq}$ KG $\succeq$ EI $\succeq$ PI $\succeq$ UCB (low β) $\succ$ DM. Finally, although TS is known to be explorative [Do et al., 2024], its exact placement in this ranking remains unclear. This ordering reflects a general quantitative understanding within the BO community; however, the relative explorative behavior of acquisition functions may vary depending on the specific problem setting. In particular, the relationship between information-theoretic acquisition functions and KG remains uncertain, as indicated by the question mark.

3.2 EXPLORATION METHODS

We introduce two quantities to evaluate the exploration behavior of different black-box optimization methods based on the locations of their observations in the search space.

Observation Traveling Salesman Distance (OTSD). OTSD quantifies the minimum Euclidean distance required to connect all observation points by formulating the problem as a traveling salesman problem (TSP). Given a set of t observation points $X_t \coloneqq \{\boldsymbol{x}_i\}_{i=1}^t$ from $\mathcal{D}_t$, OTSD is defined as the total length of the shortest possible route that visits each observation point exactly once and returns to the starting point. Mathematically, it is expressed as:

$$\mathrm{OTSD}(X_t) \coloneqq \min_{\tau \in S_t} \left(\sum_{i=1}^{t} \|\boldsymbol{x}_{\tau(i)} - \boldsymbol{x}_{\tau(i+1)}\| \right), \quad (1)$$

where τ is a permutation of $\{1, 2, \ldots, t\}$ representing a tour that visits all points with $\tau(t+1) \coloneqq \tau(1)$, $\|\cdot\|$ denotes the Euclidean distance, and S_t is the set of all possible permutations of t elements. OTSD increases monotonically with the number of observations.

Since the TSP is NP-hard, we approximate the solution using an insertion heuristic method [Rosenkrantz et al., 1974], which has a time complexity of $\mathcal{O}(dT^2)$ for T observations in d dimensions, and the worst-case tour length is guaranteed to be at most twice the optimal distance. This approach conveniently allows us to track the OTSD for each $t \leq T$. The pseudocode for calculating OTSD is in Algorithm 1.

Algorithm 1 OTSD Insertion Heuristic

Input: Observation locations $X_t = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t\}$
Output: Estimated TSP distance, OTSD(X_t)

Compute pairwise distances: $D(i,j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ for all $i, j \in \{1, \ldots, t\}, i \neq j$.
Initialize the permutation $\tau : \{1\} \to \{1\}$ that represents the (trivial) tour order on one point, $\boldsymbol{x}_1$.
For a tour on k points, define $\tau_i \coloneqq \tau(i \bmod k)$.
Initialize OTSD $\leftarrow 0$
for $k = 2$ to t **do**
    For each consecutive pair in the tour τ on points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{k-1}\}$, compute the insertion cost for placing $\boldsymbol{x}_k$ between them: $\forall i = 1, \ldots, k-1$, $\Delta C(i) \coloneqq D(\tau_i, k) + D(k, \tau_{i+1}) - D(\tau_i, \tau_{i+1})$
    Identify the insertion point $i^\star$ that minimizes $\Delta C(i)$.
    Update the permutation to τ on $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k\}$ by inserting $\boldsymbol{x}_k$ at the optimal position $i^\star$.
    OTSD $\leftarrow$ OTSD $+ \Delta C(i^\star)$.
end for
Return OTSD.

Normalized OTSD. To eliminate the influence of the problem dimensionality and ensure value consistency across different problems, we propose the *normalized OTSD*:

$$\mathrm{OTSD}_{\mathrm{norm}}(X_t) \coloneqq \frac{\mathrm{OTSD}(X_t)}{\Psi(d,t)}, \quad (2)$$

where d denotes the dimensionality of the problem and $\Psi(d,t) \coloneqq 2\sqrt{5d}\left(\frac{3t}{2}\right)^{1-1/d}$ is the upper bound derived in Proposition 3.2. Unlike OTSD, the normalized OTSD is non-monotonic; a constant value indicates that the AF does not change its behavior throughout the optimization, while higher and lower values suggest increased and decreased exploration, respectively. We use the normalized OTSD to aggregate results across different problems in Section 4.

Observation Entropy (OE). By treating observation points as samples from a random variable, we quantify the uniformity of the data distribution using empirical differential entropy. Higher entropy values indicate a more uniform spread of points, reflecting greater exploration. To estimate entropy without assuming an underlying distribution, we employ a non-parametric entropy estimator.

Several methods exist, including histogram-based [Györfi and Van der Meulen, 1987], kernel-density based [Ahmad and Lin, 1976], and nearest-neighbor based [Kozachenko and Leonenko, 1987] approaches. Among them, the nearest-neighbor-based estimator, namely the Kozachenko-Leonenko (KL) estimator, stands out for its sample efficiency and applicability in moderate-dimensional cases ($d \leq 50$), while other methods are very costly for $d \geq 10$.

We define OE using the KL estimator as follows:

$$\mathrm{OE}(X_t) \coloneqq \frac{d}{t} \sum_{i=1}^{t} \log\left(\varepsilon_i^k\right) + \psi(t) - \psi(1) + \log V_d, \quad (3)$$

where ε_i^k denotes the distance between $\boldsymbol{x}_i$ and its k-th nearest neighbor, $V_d \coloneqq \pi^{d/2}/\Gamma(1 + d/2)$ is the volume of the d-dimensional unit ball, $\Gamma(\cdot)$ denotes the Gamma function, and $\psi(t) \coloneqq \frac{\partial}{\partial t} \log \Gamma(t)$ is the digamma function. Following the recommendation of Berrett et al. [2019], we set $k = \log(t)$ (using the natural logarithm), as increasing k with t improves the efficiency of the estimation.

OE is non-monotonic and may take on either positive or negative values. An increasing OE over BO iterations indicates that the observed samples are more uniformly distributed, suggesting more exploration. Because of its non-monotonic nature, a sharp change in OE signals that an AF is altering its level of exploration over iterations. The computational complexity of OE is $\mathcal{O}(dT^2)$ for T observations in d dimensions, or $\mathcal{O}(dTk)$ if distances are provided or $t \leq T$ gets updated sequentially. Algorithm 2 details the steps for computing OE.

Algorithm 2 Kozachenko-Leonenko Entropy Estimation

Require: Observation locations $X_t = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t\}$
Ensure: Estimated observation entropy $\mathrm{OE}(X_t)$

1: Compute pairwise distances: $D(i,j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ for all $i, j \in \{1, \ldots, t\}$ with $i \neq j$.
2: For each point $\boldsymbol{x}_i$, determine ε_i^k as the k-th smallest value in the set $\{D(i,j) : j \neq i\}$.
3: Compute the volume of the unit ball in $\mathbb{R}^d$: $V_d = \frac{\pi^{d/2}}{\Gamma\left(1+\frac{d}{2}\right)}$, where $\Gamma(\cdot)$ denotes the gamma function.
4: Evaluate the digamma functions $\psi(t)$ and $\psi(1)$, then compute the KL entropy estimator as given in Equation (3).
5: **Return** $\mathrm{OE}(X_t)$.

We illustrate OTSD, normalized OTSD, and OE for various AFs on the 6-dimensional `Hartmann` function in Figure 2. The left panel shows the OTSD, which increases monotonically as new observations are added. The center panel presents the normalized OTSD, providing a clearer distinction among the different AFs. The right panel displays the OE results. Since DM yields very low OE values (around -134), we exclude DM from the OE plot for better visualization. Overall, the figure demonstrates the explorative behavior of the various AFs and cross-validates the performance of OTSD and OE, as both yield consistent rankings.

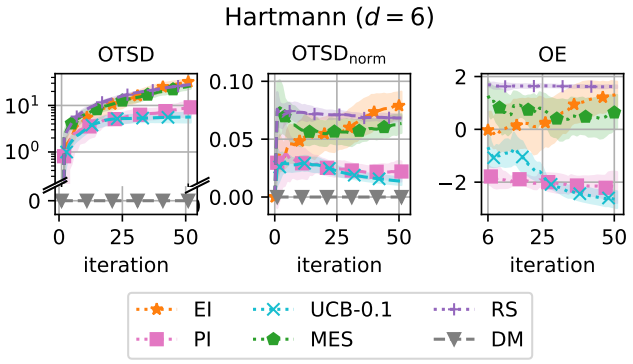


Figure 2: The exploration quantities OTSD and OE of RS, UCB ($\beta = 0.1$), EI, PI, MES, and deterministic selection (DM) on the 6-dimensional `Hartmann` function.From left to right, these plots show OTSD, normalized OTSD, and OE, respectively. DM values for OE (around -134) are hidden for better visualization. The shaded areas show the standard error of the mean.

The values of OTSD and OE are not directly comparable between different objective functions, as variations in input dimension, domain, and function landscape can significantly impact these quantities; neither quantity is invariant under a change of variables (for OE, this reflects that differential entropy is not invariant, unlike discrete entropy).

3.3 EXPLORATION BOUNDS

In Figure 2, we denote OTSD and OE as the statistical estimators based on the given input X_t. In this section, we establish bounds on their true values, denoted as $\mathrm{OTSD}^*(t)$ and $\mathrm{OE}^*(t)$ which depend only on t. The proofs for Propositions 3.2 and 3.3 are detailed in Appendices A.1 and A.2, respectively.

Proposition 3.2 (Upper and Lower Bounds for the True OTSD)**.** *Let t denote the number of observations drawn from the unit cube $[0,1]^d$ in d-dimensional space ($d \geq 3$). Building on the bounds derived in Bollobás and Meir [1992] and Balogh et al. [2024], the true observation traveling salesman distance* $\mathrm{OTSD}^*(t)$ *satisfies*

$$0 \leq \mathrm{OTSD}^*(t) \leq \Psi(d,t) \overset{def}{=} 2\sqrt{5d}\left(\frac{3}{2}t\right)^{1-1/d}. \quad (4)$$

Note that the minimum traveling salesman distance is zero, which occurs when all points coincide.

Proposition 3.3 (Upper Bound for the True OE)**.** *For any probability distribution supported in the unit cube of dimension d, the uniform distribution achieves the maximum*

differential entropy with a zero value. Conversely, the differential entropy can be made arbitrarily negative, implying that there is no finite lower bound.

Typically, the sample set X_t does not consist of independently and identically distributed points, as they are chosen adaptively via the Bayesian optimization process. Consequently, it is not necessarily the case that $\mathrm{OE}(X_t) \leq 0$, even when the samples lie within the unit hypercube. Further discussion is provided in Appendix A.3.

Extension to Non-Euclidean Domains. In many practical problems, the input domain may be non-Euclidean. Examples include integers, ordinals, categoricals, protein sequences, strings, and graphs. Quantifying exploration on these non-Euclidean domains is particularly interesting. The OTSD can still be applied in these settings provided a suitable metric is available since the insertion heuristic in Algorithm 1 relies on the triangle inequality and thus only requires a metric space. However, the estimation of OE becomes more challenging. In particular, the KL estimator in Equation (3) is designed for estimating differential entropy in Euclidean spaces. Defining an appropriate notion of entropy and developing an estimator for such irregular spaces is a case-by-case problem that requires further investigation and is beyond the scope of this paper.

4 EXPERIMENTS

We evaluate OTSD (including normalized OTSD) and OE on a wide range of synthetic and real-world benchmarks to tackle the following three research questions:

RQ1. Are the proposed quantities consistent with the literature, i.e., do OTSD and OE show higher exploration levels for AFs that are known to be more explorative?

RQ2. How do AFs, whose exploration level has not yet been discussed, relate to others in terms of exploration?

RQ3. What is the relationship between the level of exploration and optimization performance?

4.1 EXPERIMENTAL SETUP

Evaluation. For each run of an AF on a given benchmark, we record the locations $\boldsymbol{x}_i \in \mathcal{X}$ and their corresponding function values $y_{\boldsymbol{x}_i} \in \mathbb{R}$. From these observations, we compute the OTSD, the OE, and the performance, defined as the highest function value observed during the run. To aggregate OTSD results across different problems, we normalize OTSD using Equation (2) and then average these normalized values across the selected benchmarks. We give runtimes for OTSD and OE in Appendix E; OTSD is fast to compute, requiring less than 200 ms while OE needs ≈ 5 s for 1,000 observations in a $20d$ space. We also empirically validate Proposition 3.2 in Appendix C.

Since there is no straightforward method to normalize OE and performance across problems with varying dimensions, we assess their relative rankings of competing methods. Specifically, we compute the mean OE or performance for each method on each problem over ten repetitions, rank these mean values per problem, and then average the rankings across problems to obtain the *mean relative ranking* (individual optimization performances for each benchmark are provided in Appendix D.6). Note that for OE, the ranking is reversed so that more explorative methods receive a larger rank, ensuring consistency with the normalized OTSD results. Because the KL estimator exhibits significant bias in high dimensions, we report OE only for low-dimensional experiments ($d \leq 20$); for high-dimensional real-world experiments, we exclusively use OTSD.

For better clarity, we only plot the mean values for normalized OTSD and the ranks. In Appendix D.2, we show the figures with the standard error of the mean.

We also compare against the star discrepancy [Niederreiter, 1992] as a measure of uniformity in the input space using the parallel implementation by Clément et al. [2023]. Since computing the star discrepancy quickly becomes infeasible for high-dimensional problems or long sequences, we only report results for the synthetic, low-dimensional benchmarks with at most 8 dimensions and 200 function evaluations. Importantly, low star discrepancy values indicate a more uniform distribution of points, i.e., higher exploration. Therefore, the star discrepancy is inversely related to OTSD and OE.

Acquisition Functions. We study the following AFs: expected improvement (EI), probability of improvement (PI), max-value entropy search (MES), Thompson sampling (TS), and upper confidence bound (UCB). We also include knowledge gradient (KG) in our comparison but only apply it to low-dimensional problems ($d \leq 10$) due to its high computational cost. We further compare with the popular `CMA-ES` [Hansen, 2016], which is an evolutionary method for gradient-free continuous non-convex optimization using the implementation by Hansen et al. [2019], version 3.3.0.

In addition to the simple BO setup, we experiment with two common techniques for high-dimensional Bayesian optimization (HDBO): TRs and RAASP, both introduced in the `TuRBO` algorithm [Eriksson et al., 2019]. In `TuRBO`, TRs, TS, and RAASP are interwoven and not studied independently. To allow assessing their individual effects, we therefore consider adaptations of these techniques.

Finally, we study the effect of batched evaluations. In particular, we use the batched variants of the aforementioned AFs where available with batch sizes of $q = 8$ and $q = 32$. While it is not uncommon to study the number of *batch*

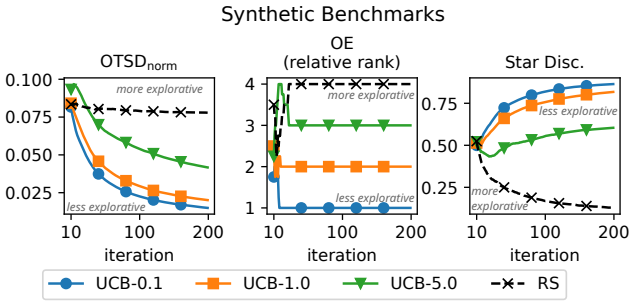


Figure 3: Normalized OTSD and OE rank averaged across all the synthetic benchmarks. A lower rank means lower exploration. We do not show the initial design of experiments phase.

evaluations to assess the effect of batching, we always study the number of function evaluations as we are interested in how batching changes exploration, i.e., we count one batch of size q as q individual function evaluations.

Benchmarks. We evaluate several AFs and variations thereof on nine benchmark problems, ranging from low-dimensional synthetic problems to noisy, high-dimensional simulations. A summary of the benchmarks is given in Table 1 in Appendix B. The $2d$ Branin, $4d$ Levy, $6d$ Hartmann, and $8d$ Griewank problems are from Surjanovic and Bingham, the $8d$ Lasso-Diabetes and $180d$ Lasso-DNA problems from Šehić et al. [2022], the $60d$ Rover and $14d$ Robot Pushing problems from Wang et al. [2018], and the $124d$ Mopta08 problem from Eriksson and Jankowiak [2021].

4.2 EMPIRICAL VALIDATION OF OTSD AND OE

We begin by empirically validating that OTSD and OE effectively quantify exploration by applying them to methods that exhibit increasing levels of explorative behavior. Figure 3 shows the normalized OTSD (left) and the mean OE ranks (right) averaged over the synthetic benchmarks. As expected, UCB with a low $\beta = 0.1$ (blue) achieves the lowest normalized OTSD and mean OE rank, followed by UCB with a moderate $\beta = 1$ (orange). UCB with a high $\beta = 5$ (green) achieves the highest OTSD and OE. The OTSD curves going down indicate that, as expected, the AFs become more exploitative over time. Comparing OTSD and OE to the star discrepancy, we see that both are strongly correlated with the star discrepancy. Importantly, they come at a considerably lower computational cost, especially in high dimensions, with the normalized OTSD being computed in a few hundred milliseconds, while the star discrepancy requires several minutes to hours to compute for high-dimensional problems or long sequences. The strong correlation between the star discrepancy and the normalized OTSD is a recurring theme in our experiments. Appendix D.1 shows the same result for real-world benchmarks; we also show the same analysis for `CMA-ES` with varying levels of σ_0.

4.3 SYNTHETIC BENCHMARKS

Figure 5 shows the performance of the various AF (without RAASP sampling and TRs), averaged over the four synthetic benchmarks. We observe that at the start of the optimization, right after the design of experiments (DoE) phase, PI is the least explorative AF as it has low normalized OTSD and the highest OE relative rank of all AFs. UCB-1.0 is similarly unexplorative, starting only slightly more explorative than PI and ending up overtaking PI as the least explorative AF. In contrast, TS starts as the most explorative AF according to both OTSD and OE. Eventually, MES, KG, and `CMA-ES` overtake TS as the most explorative AF. EI is on the same level of exploration as TS. TS and UCB-1 shows the best relative rank optimization performance.

Next, we study the effect of batching, TR, and RAASP sampling on the level of exploration for EI in Figure 6; see Appendix D.5 for the same setting on other AFs. Batching increases exploration: EI with a batch size of 32 (red) is the most explorative variant as indicated by both OTSD and OE. The variant with the smaller batch size 8 (purple) is the next most explorative variant, followed by regular EI. RAASP and TRs, on the other hand, lower the level of exploration. Both RAASP and TRs get a similarly low OE rank, indicating that they are similarly unexplorative. However, while RAASP is the highest performing variant (orange), TRs reduce exploration on a similar level as RAASP, but it is one of the worst variants, as shown in the right panel of Figure 6. Batching also degrades optimization performance, as is expected when plotting against the number of function evaluations.

4.4 REAL-WORLD BENCHMARKS

Figure 7 shows the performance of the various AF (without RAASP sampling and TRs) on the real-world benchmarks. Since most real-world benchmarks are in high dimensions, we do not plot OE for the aggregated results. The exploration behavior in high dimensions is highly consistent, with minimal overlap between the OTSD curves. We hypothesize that EI and UCB-1.0 show the best optimization performance due to their balanced the EETO as shown by the OTSD. The overly-explorative TS shows the worst performance, followed by the less-explorative PI. Compared to the other methods, MES and TS show a unique behavior: MES initially becomes less explorative, indicated by a decreasing normalized OTSD, but then reverses this trend and becomes more explorative. TS shows the opposite behavior. Initially, it strives for pure exploration and is even more explorative than RS. We explain this behavior with TS choosing areas where the surrogate exhibits high posterior variance, choosing points distant to previous observations

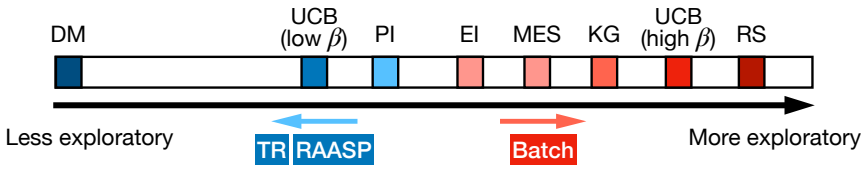


Figure 4: Empirical AF (and their variants) exploration taxonomy based on the quantitative exploration methods OTSD and OE. It represents a general understanding and may differ in specific problems.

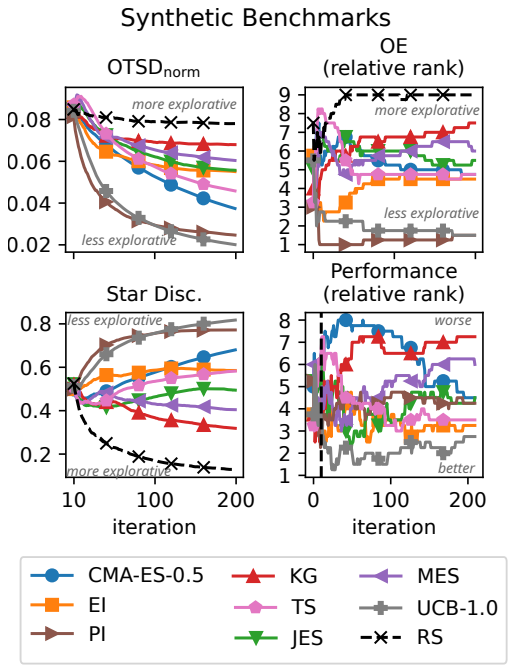


Figure 5: Normalized OTSD, average rank for OE and optimization performance on the synthetic benchmarks.

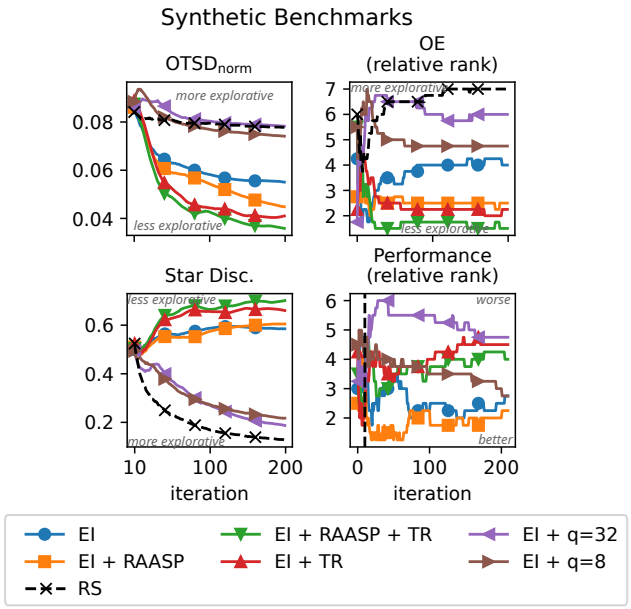


Figure 6: Normalized OTSD, average ranks for OE and optimization performance for EI and its variations on the synthetic benchmarks.

and hence yielding high OTSD. Later on, as the posterior variance of the surrogate decreases, TS becomes less explorative.

We also study the effect of TRs, RAASP, and batching on the real-world benchmarks. The results are similar to the synthetic case, so we save them for Appendix D.3.

Impact of the Dimensionality. In addition to distinguishing between synthetic and real-world scenarios, we examine OTSD and OE across low- and high-dimensional regimes. Two key differences emerge: first, TS is considerably more explorative in high-dimensional benchmarks; second, while EI is more explorative than other methods in low dimensions, it becomes less so in high dimensions. One potential explanation is that in high-dimensional spaces, the GP may struggle to accurately learn the objective function, further encouraging explorative sampling, which significantly impacts TS. Moreover, lengthscales are underestimated in higher dimensions, causing EI to make more conservative predictions than in low-dimensional settings. Due to space limitations, we present their synthetic problem results in Appendix D.4.

4.5 EXPLORATION TAXONOMY

Our empirical results on synthetic and real-world benchmarks confirm the partial ordering KG $\succeq$ EI $\succeq$ PI, which was common knowledge within the community as discussed in Section 3.1. Furthermore, our findings reveal that KG is slightly more explorative than MES, addressing a missing link in previous studies. TS is challenging to classify because its behavior varies dramatically between low and high dimensions. In low dimensions, TS performs similarly to EI; while it becomes overly explorative in high dimensions – surpassing even RS in terms of normalized OTSD. Expanding the exploration taxonomy, we empirically demonstrate that techniques such as trust regions (TRs) and RAASP consistently reduce exploration, whereas batching tends to increase exploration. Finally, our findings indicate that the best-performing methods tend not to exhibit extreme exploration quantities; their OTSD and OE values are neither

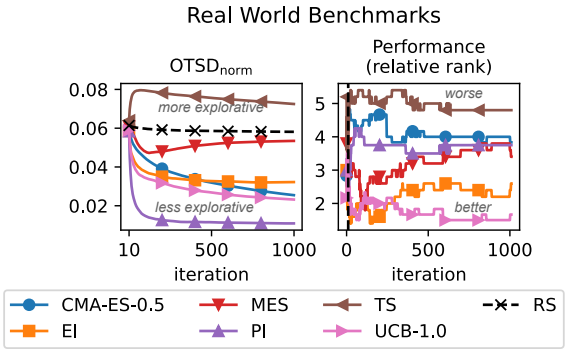


Figure 7: Normalized OTSD and average optimization performance rank on the real-world benchmarks.

excessively high nor excessively low compared to other approaches. However, they are often on the less explorative side of the spectrum, as demonstrated by the decrease in OTSD during the optimization iterations. To summarize our results, we present a revised empirical AF exploration taxonomy in Figure 4 based on our new OTSD and OE quantities.

5 CONCLUSION

In this work, we introduce two novel methods and their theoretical bounds to quantify the magnitude of exploration in various acquisition functions. We conduct extensive experiments on synthetic and real-world benchmarks, spanning low- and high-dimensional settings, to evaluate commonly used acquisition functions and their variants. Our empirical findings demonstrate that OTSD and OE effectively capture the level of exploration exhibited by these acquisition functions. Finally, we present the first empirical taxonomy of acquisition function exploration based on these quantification methods. These results offer valuable insights for designing new acquisition functions, constructing acquisition function portfolios, or controlling the optimization. For instance, an AF having higher OTSD than a random search (see TS in Figure 7) can serve as a warning sign that this AF is too explorative for the problem at hand. In such situations, one could either switch to more local AF according to the taxonomy or combine the AF with RAASP sampling. Similarly, if an AF approaches the OTSD of a random search after being less explorative at the beginning of the optimization (see MES in Figure 7), it could be an early-stopping indicator where the AF exhaustively visited all local minima and the optimization can come to an end.

Limitations and Future Work. While our results are obtained by extensive empirical experimentation, they do not consider the effect of GP hyperparameters and hyperpriors on the behavior of AFs and are limited to continuous domains. In the future, we will expand the taxonomy to include additional AFs, for instance, other information-theoretic AFs [Hennig and Schuler, 2012, Hernández-Lobato et al., 2014, Hvarfner et al., 2022, Cheng et al., 2025], and study the effect of kernel and likelihood functions and their hyperparameters on the behavior of AFs in Bayesian optimization. Furthermore, we will study if OTSD and OE can be extended to other domains, such as non-Euclidean spaces.

Acknowledgements

This project was partly supported by the Wallenberg AI, Autonomous Systems, and Software program (WASP) funded by the Knut and Alice Wallenberg Foundation, the AFOSR awards FA9550-20-1-0138, with Dr. Fariba Fahroo as the program manager, DOE award DE-SC0023346, and by the US Department of Energy's Wind Energy Technologies Office. The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725

References

Salem F Adra and Peter J Fleming. Diversity management in evolutionary many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 15(2):183–195, 2010.

Ibrahim Ahmad and Pi-Erh Lin. A nonparametric estimation of the entropy for absolutely continuous distributions (corresp.). *IEEE Transactions on Information Theory*, 22 (3):372–375, 1976.

Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *International conference on machine learning*, pages 151–160. PMLR, 2019.

Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *Advances in Neural Information Processing Systems*, 33:21524–21538, 2020.

József Balogh, Felix Christian Clemen, and Adrian Dumitrescu. On a traveling salesman problem for points in the unit cube. *Algorithmica*, 86(9):3054–3078, 2024.

Carolin Benjamins, Elena Raponi, Anja Jankovic, Koen van der Blom, Maria Laura Santoni, Marius Lindauer, and Carola Doerr. PI is back! Switching Acquisition Functions in Bayesian Optimization. In *2022 NeurIPS Workshop on Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems*, 2022.

Thomas B Berrett, Richard J Samworth, and Ming Yuan. Efficient multivariate entropy estimation via k-nearest

neighbour distances. *The Annals of Statistics*, 47(1):288–318, 2019.

Hildo Bijl, Thomas B Schön, Jan-Willem van Wingerden, and Michel Verhaegen. A sequential Monte Carlo approach to Thompson sampling for Bayesian optimization. *arXiv preprint arXiv:1604.00169*, 2016.

Bernd Bischl, Simon Wessing, Nadja Bauer, Klaus Friedrichs, and Claus Weihs. MOI-MBO: multiobjective infill for parallel model-based optimization. In *Learning and Intelligent Optimization: 8th International Conference, Lion 8, Gainesville, FL, USA, February 16-21, 2014. Revised Selected Papers 8*, pages 173–186. Springer, 2014.

Béla Bollobás and Amram Meir. A travelling salesman problem in the k-dimensional unit cube. *Operations research letters*, 11(1):19–21, 1992.

Edmund Burke, Steven Gustafson, Graham Kendall, and Natalio Krasnogor. Advanced population diversity measures in genetic programming. In *Parallel Problem Solving from Nature—PPSN VII: 7th International Conference Granada, Spain, September 7–11, 2002 Proceedings 7*, pages 341–350. Springer, 2002.

Cambridge. Cambridge online dictionary. `https://dictionary.cambridge.org/us/dictionary/english/exploration`. Accessed: 2025-02-08.

Nachol Chaiyaratana, Theera Piroonratana, and Nuntapon Sangkawelert. Effects of diversity control in single-objective and multi-objective genetic algorithms. *Journal of Heuristics*, 13:1–34, 2007.

Nuojin Cheng, Leonard Papenmeier, Stephen Becker, and Luigi Nardi. A unified framework for entropy search and expected improvement in Bayesian optimization. *arXiv preprint arXiv:2501.18756*, 2025.

François Clément, Diederick Vermetten, Jacob De Nobel, Alexandre D Jesus, Luís Paquete, and Carola Doerr. Computing star discrepancies with numerical black-box optimization algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1330–1338, 2023.

Matej Črepinšek, Marjan Mernik, and Shih-Hsi Liu. Analysis of exploration and exploitation in evolutionary algorithms by ancestry trees. *International Journal of Innovative Computing and Applications*, 3(1):11–19, 2011.

Matej Črepinšek, Shih-Hsi Liu, and Marjan Mernik. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)*, 45(3):1–33, 2013.

George De Ath, Richard M Everson, Alma AM Rahat, and Jonathan E Fieldsend. Greed is Good: Exploration and Exploitation Trade-offs in Bayesian Optimisation. *ACM Transactions on Evolutionary Learning and Optimization*, 1(1):1–22, 2021.

Sylvain Delattre and Nicolas Fournier. On the Kozachenko-Leonenko entropy estimator. *Journal of Statistical Planning and Inference*, 185:69–93, 2017.

Luc Devroye and László Györfi. On the consistency of the Kozachenko-Leonenko entropy estimate. *IEEE Transactions on Information Theory*, 68(2):1178–1185, 2021.

Bach Do, Taiwo Adebiyi, and Ruda Zhang. Epsilon-greedy Thompson sampling to Bayesian optimization. *Journal of Computing and Information Science in Engineering*, 24(12), 2024.

Agoston E Eiben and James E Smith. *Introduction to evolutionary computing*. Springer, 2015.

David Eriksson. Trust Region Bayesian Optimization (TuRBO). BoTorch Tutorials, 2025. URL `https://botorch.org/docs/tutorials/turbo_1/`. Accessed: 2025-02-06.

David Eriksson and Martin Jankowiak. High-Dimensional Bayesian Optimization with Sparse Axis-Aligned Subspaces. In *Uncertainty in Artificial Intelligence*, pages 493–503. PMLR, 2021.

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable Global Optimization via Local Bayesian Optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhiwei Feng, Qingbin Zhang, Qingfu Zhang, Qiangang Tang, Tao Yang, and Yang Ma. A multiobjective optimization based framework to balance the global exploration and local exploitation in expensive optimization. *Journal of Global Optimization*, 61:677–694, 2015.

Peter I Frazier. *Knowledge-Gradient Methods for Statistical Learning*. PhD thesis, Princeton University Princeton, 2009.

David Ginsbourger and Rodolphe Le Riche. Towards Gaussian process-based optimization with finite time horizon. In *mODa 9–Advances in Model-Oriented Design and Analysis: Proceedings of the 9th International Workshop in Model-Oriented Design and Analysis held in Bertinoro, Italy, June 14-18, 2010*, pages 89–96. Springer, 2010.

Javier González, Zhenwen Dai, Philipp Hennig, and Neil Lawrence. Batch Bayesian optimization via local penalization. In *Artificial intelligence and statistics*, pages 648–657. PMLR, 2016.

László Györfi and Edward C Van der Meulen. Density-free convergence properties of various estimators of entropy. *Computational Statistics & Data Analysis*, 5(4):425–436, 1987.

John H Halton. Algorithm 247: Radical-inverse quasi-random point sequence. *Communications of the ACM*, 7 (12):701–702, 1964.

Nikolaus Hansen. The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

Nikolaus Hansen, Youhei Akimoto, and Petr Baudis. CMA-ES/pycma on Github. Zenodo, DOI:10.5281/zenodo.2559634, February 2019. URL `https://doi.org/10.5281/zenodo.2559634`.

Philipp Hennig and Christian J Schuler. Entropy Search for Information-Efficient Global Optimization. *Journal of Machine Learning Research*, 13(6), 2012.

José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive Entropy Search for Efficient Global Optimization of Black-box Functions. *Advances in Neural Information Processing Systems*, 27, 2014.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

Carl Hvarfner, Frank Hutter, and Luigi Nardi. Joint Entropy Search for Maximally-Informed Bayesian Optimization. *Advances in Neural Information Processing Systems*, 35: 11494–11506, 2022.

Carl Hvarfner, Erik Orm Hellsten, and Luigi Nardi. Vanilla Bayesian Optimization Performs Great in High Dimensions. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 20793–20817. PMLR, 21–27 Jul 2024.

Kazuyuki Inoue, Taku Hasegawa, Naoki Mori, and Keinosuke Matsumoto. Analyzing exploration exploitation trade-off by means of PI similarity index and dictyostelium based genetic algorithm. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 2548–2555. IEEE, 2015.

Jernej Jerebic, Marjan Mernik, Shih-Hsi Liu, Miha Ravber, Mihael Baketarić, Luka Mernik, and Matej Črepinšek. A novel direct measure of exploration and exploitation based on attraction basins. *Expert Systems with Applications*, 167:114353, 2021.

Shali Jiang, Daniel Jiang, Maximilian Balandat, Brian Karrer, Jacob Gardner, and Roman Garnett. Efficient nonmyopic Bayesian optimization via one-shot multi-step trees. *Advances in Neural Information Processing Systems*, 33: 18039–18049, 2020.

Donald R Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21:345–383, 2001.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13:455–492, 1998.

Lyudmyla F Kozachenko and Nikolai N Leonenko. Sample estimate of the entropy of a random vector. *Problems of Information Transmission*, 23:9–16, 1987.

Volodymyr Kuleshov and Doina Precup. Algorithms for multi-armed bandit problems. *arXiv preprint arXiv:1402.6028*, 2014.

Brian McGinley, John Maher, Colm O'Riordan, and Fearghal Morgan. Maintaining healthy population diversity using adaptive crossover, mutation, and selection. *IEEE Transactions on Evolutionary Computation*, 15(5): 692–714, 2011.

Alfonsas Misevičius. Generation of grey patterns using an improved genetic evolutionary algorithm: Some new results. *Information Technology and Control*, 40(4):330–343, 2011.

Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*. SIAM, 1992.

Dávid Pál, Barnabás Póczos, and Csaba Szepesvári. Estimation of Rényi entropy and mutual information based on generalized nearest-neighbor graphs. *Advances in neural information processing systems*, 23, 2010.

Bahador Rashidi, Kerrick Johnstonbaugh, and Chao Gao. Cylindrical Thompson sampling for high-dimensional Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 3502–3510. PMLR, 2024.

Carl Edward Rasmussen, Christopher KI Williams, et al. *Gaussian Processes for Machine Learning*, volume 1. Springer, 2006.

Daniel J Rosenkrantz, Richard Edwin Stearns, and Philip M Lewis. Approximate algorithms for the traveling salesperson problem. In *15th Annual Symposium on Switching and Automata Theory (swat 1974)*, pages 33–42. IEEE, 1974.

Kenan Šehić, Alexandre Gramfort, Joseph Salmon, and Luigi Nardi. LassoBench: A High-Dimensional Hyperparameter Optimization Benchmark Suite for Lasso. In

International Conference on Automated Machine Learning, pages 2–1. PMLR, 2022.

András Sóbester, Stephen J Leary, and Andy J Keane. On the Design of Optimization Strategies Based on Global Response Surface Approximation Models. *Journal of Global Optimization*, 33:31–59, 2005.

Ilya M Sobol. The distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational mathematics and mathematical physics*, 7:86–112, 1967.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *Proceedings of the International Conference on Machine Learning*, 2010.

Yi Sun, Faustino Gomez, and Jürgen Schmidhuber. Planning to be surprised: Optimal Bayesian exploration in dynamic environments. In *Artificial General Intelligence: 4th International Conference, AGI 2011, Mountain View, CA, USA, August 3-6, 2011. Proceedings 4*, pages 41–51. Springer, 2011.

Sonja Surjanovic and Derek Bingham. Virtual library of simulation experiments: Test functions and datasets, optimization test problems. `https://www.sfu.ca/~ssurjano/optimization.html`. Accessed: 2024-09-01.

Richard S Sutton. Reinforcement learning: An introduction. *A Bradford Book*, 2018.

Ben Tu, Axel Gandy, Nikolas Kantas, and Behrang Shafei. Joint Entropy Search for Multi-objective Bayesian Optimization. *Advances in Neural Information Processing Systems*, 35:9922–9938, 2022.

Mikdam Turkey and Riccardo Poli. A model for analysing the collective dynamic behaviour and characterising the exploitation of population-based algorithms. *Evolutionary Computation*, 22(1):159–188, 2014.

Jialei Wang, Scott C. Clark, Eric Liu, and Peter I. Frazier. Parallel Bayesian global optimization of expensive functions. *Oper. Res.*, 68(6):1850–1865, November 2020. ISSN 0030-364X.

Zi Wang and Stefanie Jegelka. Max-value Entropy Search for Efficient Bayesian Optimization. In *International Conference on Machine Learning*, pages 3627–3635. PMLR, 2017.

Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched Large-scale Bayesian Optimization in High-dimensional Spaces. In *International Conference on Artificial Intelligence and Statistics*, pages 745–754. PMLR, 2018.

Ronald J Williams and Jing Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

James T Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Pathwise Conditioning of Gaussian Processes. *Journal of Machine Learning Research*, 22(105):1–47, 2021.

Jian Wu. *Knowledge gradient methods for Bayesian optimization*. PhD thesis, Cornell University, 2017.

Jian Wu and Peter Frazier. Practical two-step lookahead Bayesian optimization. *Advances in neural information processing systems*, 32, 2019.

Antanas Žilinskas and James Calvin. Bi-objective decision making in global optimization based on statistical models. *Journal of Global Optimization*, 74:599–609, 2019.

Appendix
(Supplementary Material)

Leonard Papenmeier[*1] **Nuojin Cheng**[*2] **Stephen Becker**[2] **Luigi Nardi**[1, 3]

[1]Department of Computer Science, Lund University, Lund, Sweden
[2]Department of Applied Mathematics, University of Colorado Boulder, Boulder, Colorado, USA
[3]DBtune

A PROOFS

A.1 OTSD UPPER BOUND

Proof. Our proof follows from Balogh et al. [2024, Theorem 1.3]. In that work, the authors show that the d-norm length of a Hamiltonian cycle with t nodes in a d-dimensional unit cube, denoted by

$$s_d^{\text{HC}}(t) \coloneqq \left(\sum_{i=1}^{t} \sqrt{\mathrm{e}_i^d} \right)^{1/d}, \tag{5}$$

satisfies

$$s_d^{\text{HC}}(t) \leq 3\sqrt{5} \left(\frac{2}{3} \right)^{1/d} \sqrt{d}, \tag{6}$$

where

$$\mathrm{e}_i \coloneqq \|\boldsymbol{x}_{\tau(i)} - \boldsymbol{x}_{\tau(i)+1}\|_2 \tag{7}$$

represents the Euclidean distance between successive nodes.

The true total distance with t nodes, denoted by $\text{OTSD}^*(t)$, is given by

$$\text{OTSD}^*(t) \coloneqq \sum_{i=1}^{t} \mathrm{e}_i. \tag{8}$$

Applying Hölder's inequality,

$$\|\boldsymbol{e}\|_1 \leq \|\boldsymbol{e}\|_d \cdot \|\mathbf{1}\|_{(1-1/d)^{-1}}, \tag{9}$$

yields

$$\text{OTSD}^*(t) = \sum_{i=1}^{t} \mathrm{e}_i \leq s_d^{\text{HC}}(t) \cdot \left(\sum_{i=1}^{t} 1 \right)^{1-1/d}. \tag{10}$$

Since $\sum_{i=1}^{t} 1 = t$, it follows that

$$\text{OTSD}^* \leq 3\sqrt{5} \left(\frac{2}{3} \right)^{1/d} t^{1-1/d} \sqrt{d} = 2\sqrt{5d} \left(\frac{3}{2} t \right)^{1-1/d}. \tag{11}$$

□

A.2 OE UPPER BOUND

We show that the distribution with maximum differential entropy in the unit cube $[0,1]^d$ is the uniform distribution.

Proof. We wish to determine the probability density function $p(\boldsymbol{x})$, for $\boldsymbol{x} \in [0,1]^d$, that maximizes the differential entropy

$$H[p] = -\int_{[0,1]^d} p(\boldsymbol{x}) \ln p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}, \tag{12}$$

subject to the normalization constraint

$$\int_{[0,1]^d} p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = 1. \tag{13}$$

To enforce this constraint, we introduce a Lagrange multiplier λ and consider the augmented functional

$$\mathcal{L}[p] = -\int_{[0,1]^d} p(\boldsymbol{x}) \ln p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} + \lambda \left(\int_{[0,1]^d} p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} - 1 \right). \tag{14}$$

We then compute the first variation $\delta\mathcal{L}$ with respect to an arbitrary variation $\delta p(\boldsymbol{x})$, yielding

$$\delta\mathcal{L} = -\int_{[0,1]^d} \delta p(\boldsymbol{x}) \Big[\ln p(\boldsymbol{x}) + 1 \Big] \, \mathrm{d}\boldsymbol{x} + \lambda \int_{[0,1]^d} \delta p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}. \tag{15}$$

For $\delta\mathcal{L}$ to vanish for all admissible variations $\delta p(\boldsymbol{x})$, the integrand must be zero:

$$-\ln p(\boldsymbol{x}) - 1 + \lambda = 0 \quad \text{for all } \boldsymbol{x} \in [0,1]^d. \tag{16}$$

Solving for $\ln p(\boldsymbol{x})$, we obtain

$$\ln p(\boldsymbol{x}) = \lambda - 1, \tag{17}$$

which implies that

$$p(\boldsymbol{x}) = e^{\lambda - 1}. \tag{18}$$

Since $p(\boldsymbol{x})$ is constant over $[0,1]^d$, we can determine the constant by imposing the normalization condition:

$$\int_{[0,1]^d} p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = e^{\lambda - 1} \cdot 1 = 1. \tag{19}$$

Thus,

$$e^{\lambda - 1} = 1 \quad \Longrightarrow \quad \lambda - 1 = 0 \quad \Longrightarrow \quad \lambda = 1, \tag{20}$$

yields

$$p(\boldsymbol{x}) = 1 \quad \text{for all } \boldsymbol{x} \in [0,1]^d. \tag{21}$$

This completes the proof that the maximum entropy distribution on the d-dimensional unit cube is indeed the uniform distribution. Therefore,

$$\mathrm{OE}^*(t) \leq H[p_{\text{unif}}] = 0. \tag{22}$$

□

A.3 KL ESTIMATION CONSISTENCY AND BIAS

Many studies have examined the estimation bias of the Kozachenko-Leonenko estimator (see Equation (3)). The original work by Kozachenko and Leonenko [1987] established the consistency of the estimator under mild conditions when $k = 1$. Furthermore, Pál et al. [2010] demonstrated both the consistency and the convergence rate of the nearest-neighbor-based estimator for Rényi entropies under the assumption that the entropy support is bounded. In addition, Delattre and Fournier [2017] extended these results to non-compactly supported densities, providing an upper bound for the bias of order $\mathcal{O}(t^{-2/d})$. A more recent study [Devroye and Györfi, 2021] presents a consistency result for the KL estimator even when the density function is not smooth. However, in our study the sample points in X_t are generally not independent and identically distributed, as they are collected via a Bayesian optimization process. Consequently, we believe that the practical performance of OE does not align with the findings of these previous studies.

B DETAILS ON THE EXPERIMENTAL SETUP

We run each AF in a basic BO setup where we first initialize the GP with ten observations that we sample uniformly at random from $\mathcal{X}$ and then start the BO loop for 200 iterations for synthetic and 1000 iterations for real-world problems. Similarly, we run `CMA-ES` with a population size of 5 with different initial step sizes $\sigma_0 \in \{0.05, 0.1, 0.5\}$. UCB allows one to specify an exploration parameter β, which we set to $\beta \in \{0.1, 1, 5\}$.

To evaluate the AFs, we use the default `SingleTaskGP` GP model provided by `BoTorch` version `0.12.0` [Balandat et al., 2020] as well as `BoTorch`'s provided methods to fit the GP and maximize the AF. In the case of TS, we sample from the GP posterior using pathwise conditioning [Wilson et al., 2021] and maximize the posterior sample using `BoTorch`'s `optimize_posterior_samples` function with 1024 initial random samples and 20 restarts of the gradient descent (GD) optimizer.

For RAASP sampling, we rely on the `sample_around_best` parameter for `BoTorch`'s AF optimizer that, with a probability of $\min\left(1, \frac{20}{d}\right)$, substitutes a parameter's current best configuration with a value sampled from a truncated Gaussian, centered on the incumbent observation. For TRs, we follow `TuRBO`'s specification for finding the TR bounds. We then configure the AF maximizer to respect these bounds, similar to Eriksson [2025] but dropping the sparse perturbations.

Table 1 summarizes the benchmarks used in this work.

Name	d	Noise	Synth.
Branin	2	✗	✓
Levy	4	✗	✓
Hartmann	6	✗	✓
Griewank	8	✗	✓
Lasso-Diabetes	8	✗	✗
Robot Pushing	14	✓	✗
Rover	60	✗	✗
Mopta08	124	✗	✗
Lasso-DNA	180	✗	✗

Table 1: Benchmark summary.

C EMPIRICAL VERIFICATION OF THE OTSD BOUND

To verify the upper bound stated in Proposition 3.2, we take advantage of the normalized OTSD defined in Equation (2) which makes the OTSD independent from the dimensionality d. If the approximation error to compute OTSD is ignored, the theoretical bound implies that $\text{OTSD}_{\text{norm}}(t)$ should remain below 1 for all t (or with approximation error considered, $\text{OTSD}_{\text{norm}}(t)$ should remain below 2 for all t).

We show $\text{OTSD}_{\text{norm}}$ for Random Search (RS) across various dimensionalities of the problem in Figure 8. As the figure illustrates, all values remain well below the theoretical threshold of 1. Given that RS represents a highly explorative scenario, this result empirically corroborates the bound in Proposition 3.2. Moreover, we observe that for higher-dimensional problems, the normalized OTSD converges to a specific constant, whereas for lower-dimensional problems, the convergence is less pronounced. This suggests that while the bound is reliable in high dimensions, it may not be as tight in low dimensions.

D ADDITIONAL EXPERIMENTS

D.1 EMPIRICAL VALIDATION OF OTSD AND OE

In this section, we study the OTSD for varying β-values for UCB in real-world settings and for `CMA-ES` with different step sizes, further supporting the adequacy of OTSD and OE for quantifying exploration.

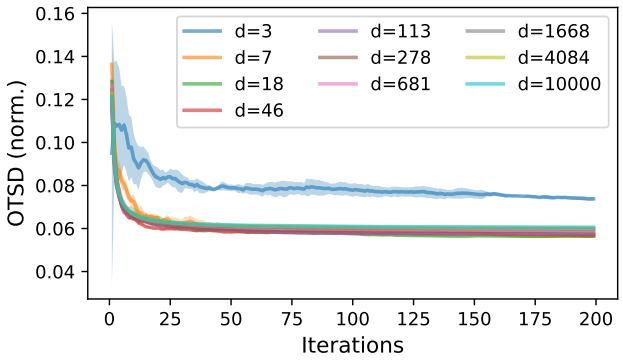
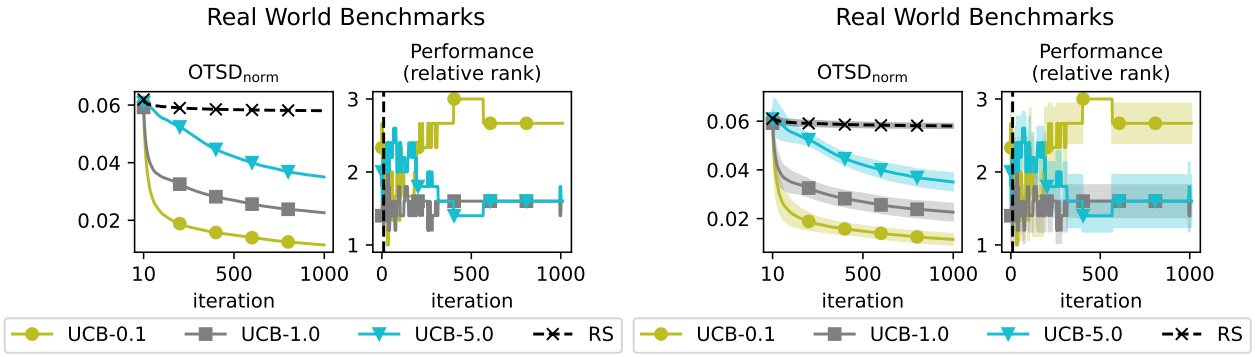


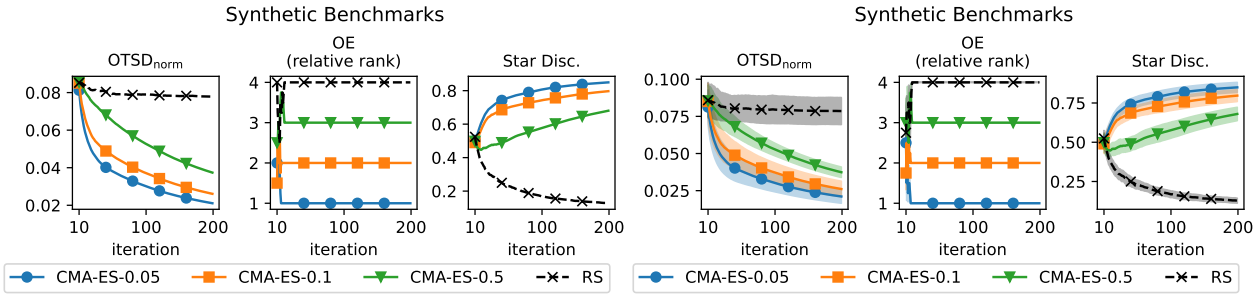
Figure 8: Normalized OTSD of Random Search for varying problem dimensions.



(a) Without error bars.

(b) With error bars.

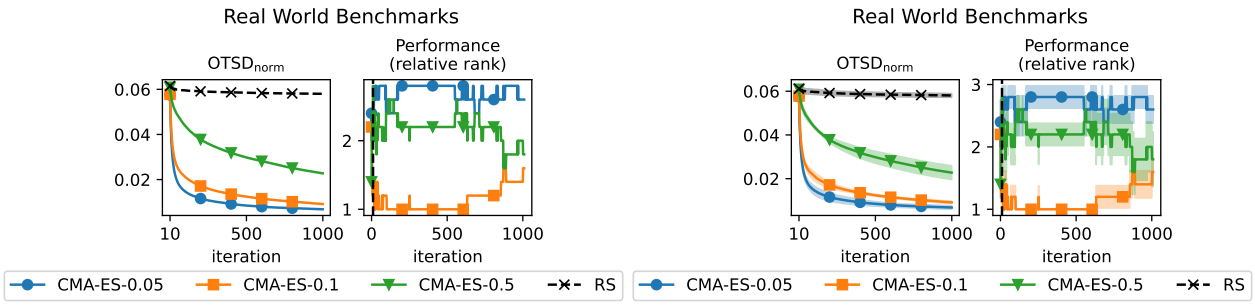
Figure 9: Normalized OTSD and mean ranks of the empirical performance for UCB with varying β-parameters on the real-world benchmarks.



(a) Without error bars.

(b) With error bars.

Figure 10: Normalized OTSD and mean ranks of the empirical performance for CMA-ES with varying σ_0-parameters on the synthetic benchmarks.



(a) Without error bars.

(b) With error bars.

Figure 11: Normalized OTSD and mean ranks of the empirical performance for `CMA-ES` with varying σ_0-parameters on the real-world benchmarks.

D.2 ERROR BARS FOR MAIN TEXT FIGURES

We report the figures from the main text with error bars indicating the standard error of the mean.

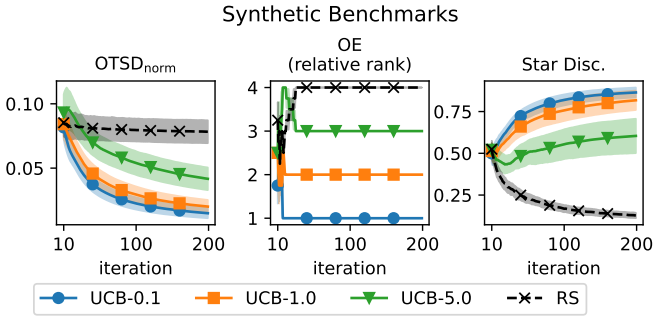


Figure 12: Normalized OTSD and OE rank averaged across all the synthetic benchmarks. A lower rank means lower exploration. We do not show the initial design of experiments phase.

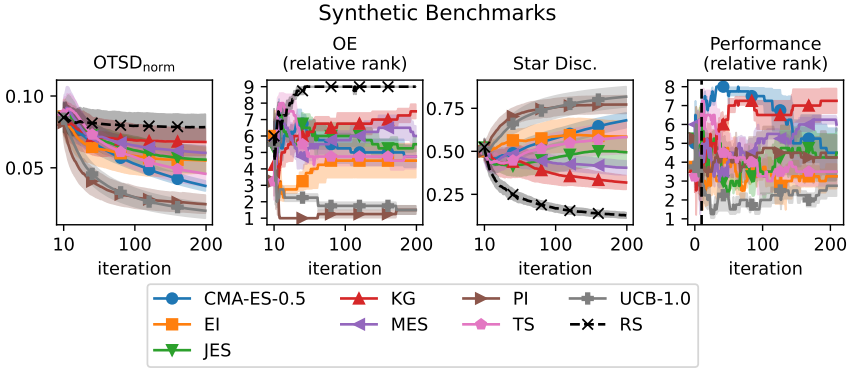


Figure 13: Normalized OTSD, average rank for OE and optimization performance on the synthetic benchmarks.

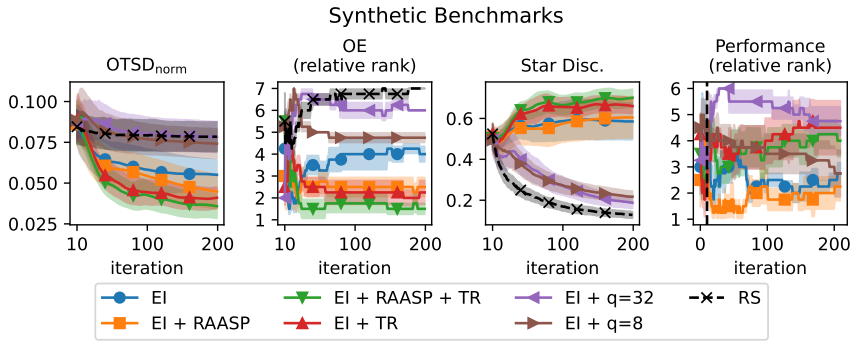


Figure 14: Normalized OTSD, average ranks for OE and optimization performance for EI and its variations on the synthetic benchmarks.

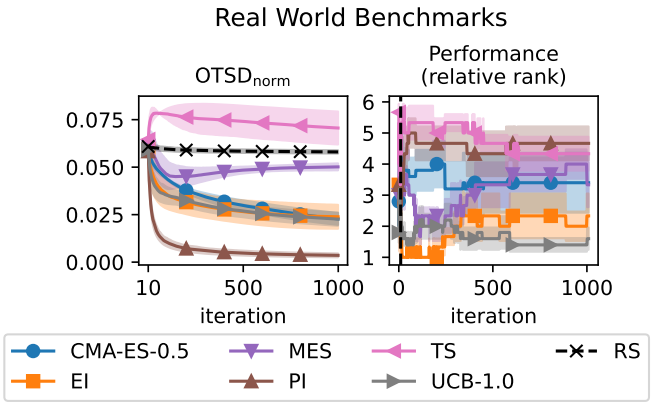
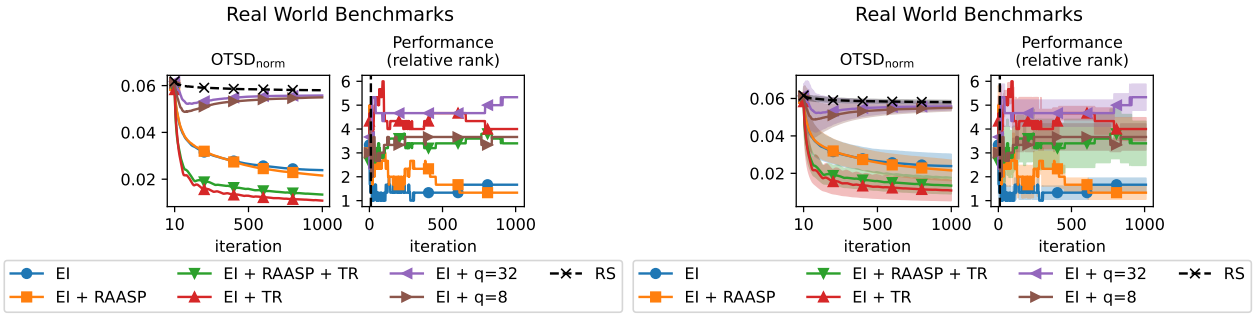


Figure 15: Normalized OTSD and average optimization performance rank on the real-world benchmarks.

D.3 TRS, RAASP, AND BATCHING ON THE REAL-WORLD BENCHMARKS



(a) Without error bars.

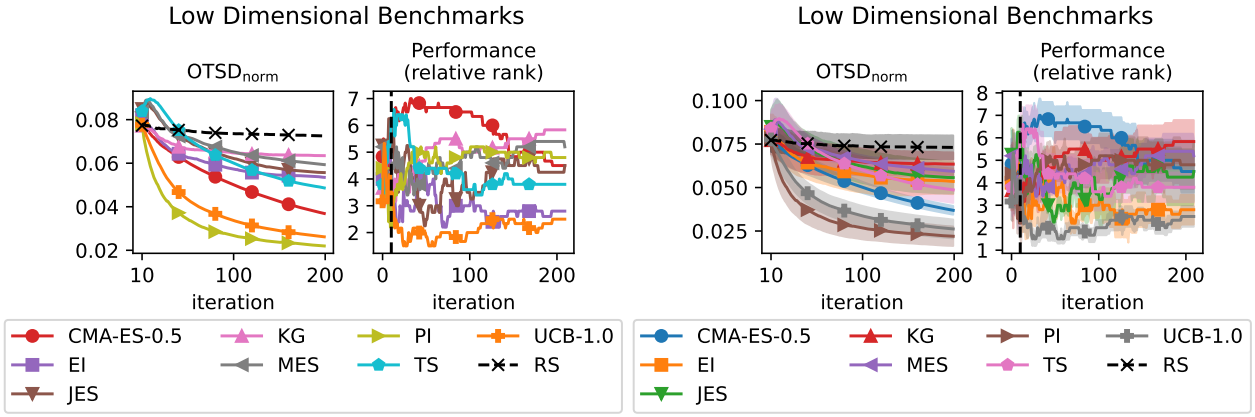
(b) With error bars.

Figure 16: Normalized OTSD and average optimization performance rank on the real-world benchmarks for EI and its variations. TRs and RAASP sampling promote exploitation.

Figure 16 zooms in on the effect of batching, TR, and RAASP sampling. Here, the behavior is similar to the synthetic benchmarks: batching increases exploration and larger batch sizes lead to more exploration while RAASP sampling and TRs reduce exploration. In particular, the level of exploitation introduced by the TRs dominates all other methods. Compared to the optimization performance, both over-exploration and over-exploitation get punished: the most and least explorative methods ('EI + TR' and 'EI + q=32') show the worst empirical performance as indicated by the high average rank of the purple and blue curves in the right panel of Figure 16.

D.4 IMPACT OF THE DIMENSIONALITY

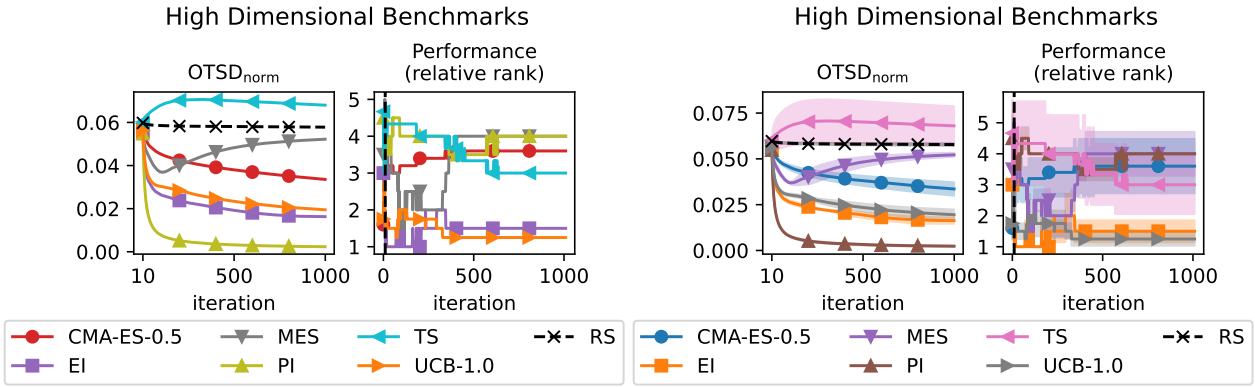
Next, we study how the dimensionality of problems affects exploration. To this end, we compare low-dimensional ($d \leq 20$, Figure 17) and high-dimensional problems ($d > 20$, Figure 18), unveiling significant differences between low- and high-dimensional regimes. While TS is eventually overtaken by MES in terms of exploration, it is by far the most explorative AF on high-dimensional problems. This is arguably due to the vast regions of high posterior uncertainty in high-dimensional spaces that allow diverse posterior samples. Similarly, MES is more explorative than other AFs (except for TS) in high-dimensional than low-dimensional spaces. Conversely, EI is considerably more explorative than UCB-1 in low-dimensional but not high-dimensional spaces. Arguably, the fast collapse of posterior uncertainty in low-dimensional spaces affects UCB-1 more than EI, making UCB almost as exploitative as PI. In both regimes, PI is most exploitative while showing mediocre to bad optimization performance. Brought together that the over-explorative TS also shows subpar optimization performance, we reinforce our conclusion that a balanced EETO is crucial for successful black-box optimizers.



(a) Without error bars.

(b) With error bars.

Figure 17: Normalized OTSD and optimization performance ranks on the low-dimensional problems.



(a) Without error bars.

(b) With error bars.

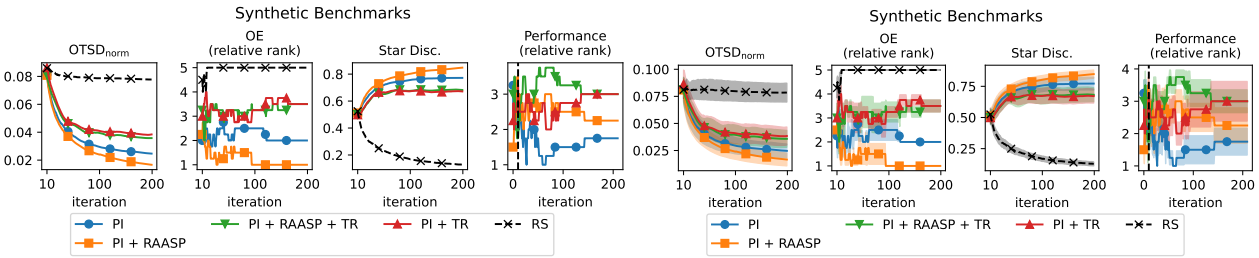
Figure 18: Normalized OTSD and optimization performance rank on the high-dimensional problems.

D.5 OPTIMIZER VARIATIONS ON DIFFERENT ACQUISITION FUNCTIONS

Here, we study the effect of RAASP sampling, TRs, and batching on all AFs used in Section 4.

D.5.1 Probability of Improvement

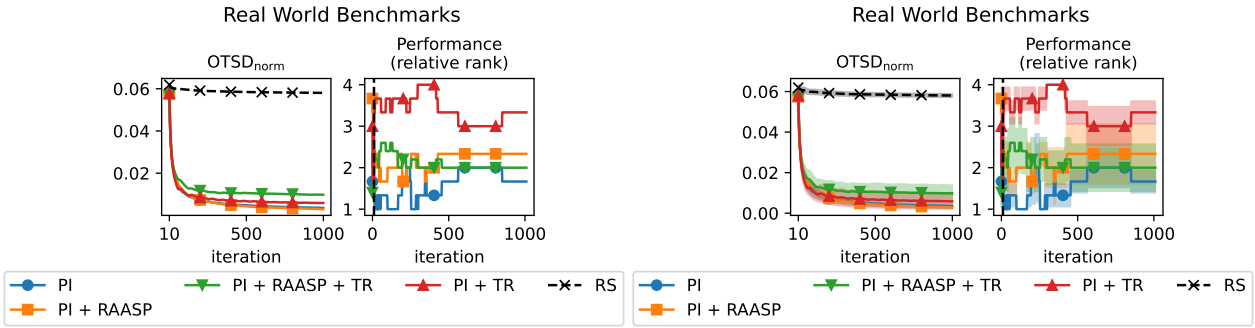
For PI, we did not study batching as it is not implemented in `BoTorch`, so we only focus on TRs and RAASP sampling.



(a) Without error bars.

(b) With error bars.

Figure 19: Effect of TRs and RAASP sampling on PI in the context of synthetic benchmarks: Both methods reduce the level of exploration.

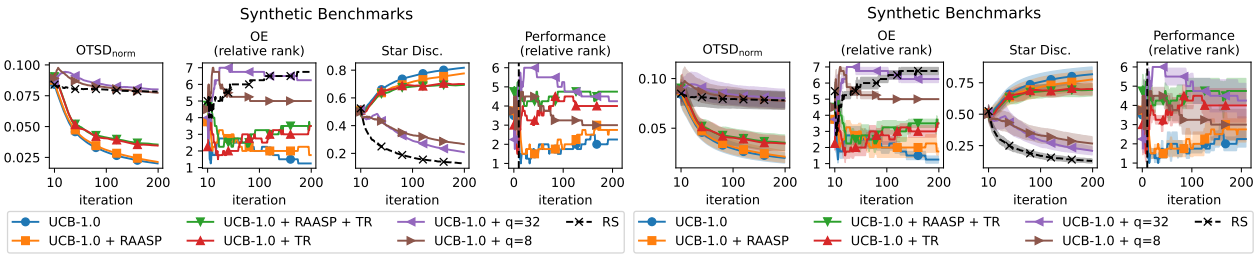


(a) Without error bars.

(b) With error bars.

Figure 20: Effect of TRs and RAASP sampling on PI in the context of real-world benchmarks: Both methods reduce the level of exploration.

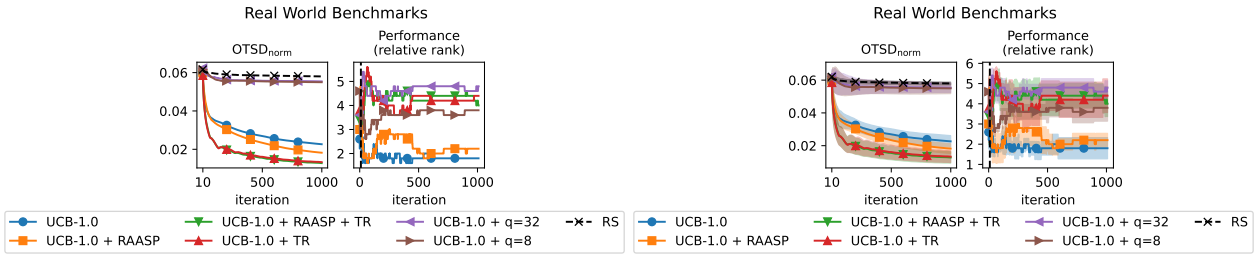
D.5.2 Upper Confidence Bounds



(a) Without error bars.

(b) With error bars.

Figure 21: Effect of TRs, RAASP sampling, and batching on UCB-1 in the context of synthetic benchmarks:RAASP sampling and TRs reduce exploration, batching increases it.

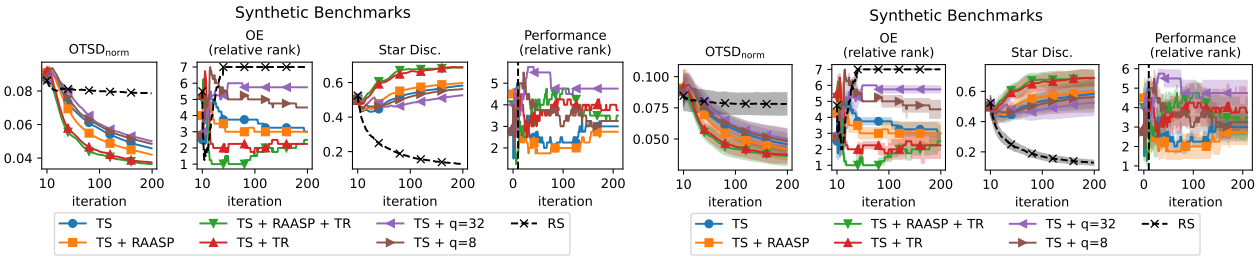


(a) Without error bars.

(b) With error bars.

Figure 22: Effect of TRs, RAASP sampling, and batching on UCB-1 in the context of real-world benchmarks:RAASP sampling and TRs reduce exploration, batching increases it.

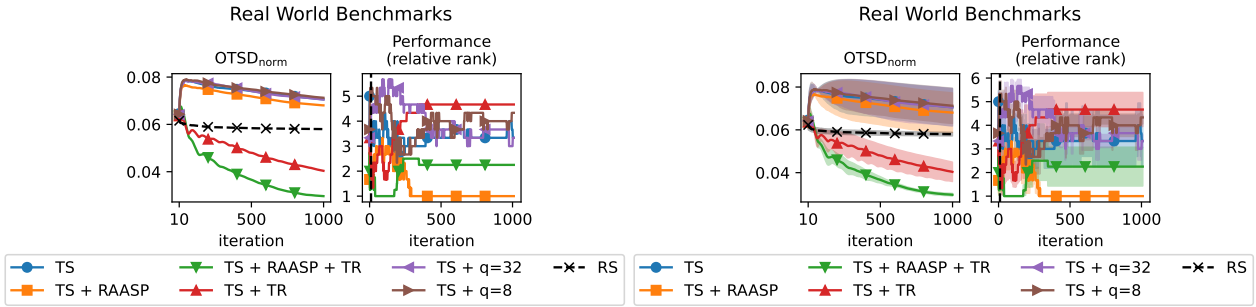
D.5.3 Thompson Sampling



(a) Without error bars.

(b) With error bars.

Figure 23: Effect of TRs, RAASP sampling, and batching on TS in the context of synthetic benchmarks:RAASP sampling and TRs reduce exploration, batching increases it. RAASP sampling also improves optimization performance.



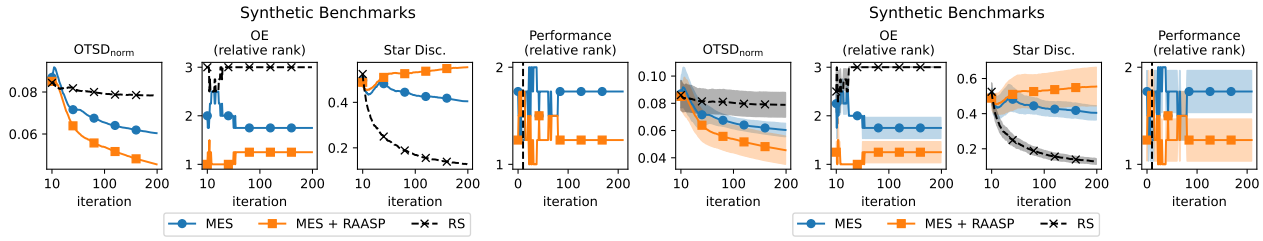
(a) Without error bars.

(b) With error bars.

Figure 24: Effect of TRs, RAASP sampling, and batching on TS in the context of real-world benchmarks:RAASP sampling and TRs reduce exploration, batching increases it. RAASP sampling also improves optimization performance.

D.5.4 Max-Value Entropy Search

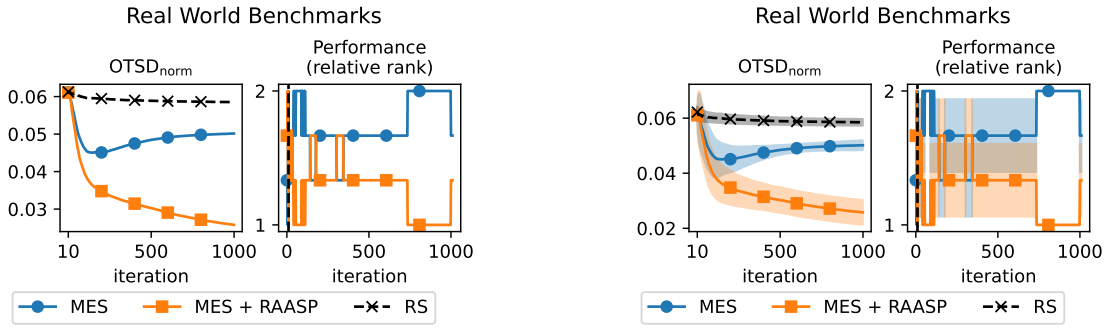
For MES, we only study the effect of RAASP sampling as we observed model-fitting errors for the TRs. Furthermore, batching is not implemented for MES in `BoTorch`.



(a) Without error bars.

(b) With error bars.

Figure 25: Effect of RAASP sampling on MES in the context of synthetic benchmarks: RAASP sampling reduces the level of exploration.



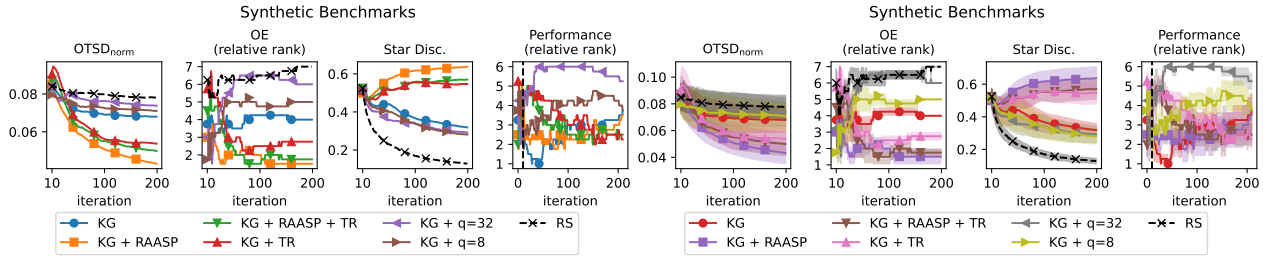
(a) Without error bars.

(b) With error bars.

Figure 26: Effect of RAASP sampling on MES in the context of real-world benchmarks: RAASP sampling reduces the level of exploration.

D.5.5 Knowledge Gradient

We only ran KG for low-dimensional synthetic benchmarks due to its high computational cost in high dimensions.



(a) Without error bars.

(b) With error bars.

Figure 27: Effect of TRs, RAASP sampling, and batching on KG in the context of synthetic benchmarks:RAASP sampling and TRs reduce exploration, batching increases it.

D.6 OPTIMIZATION PERFORMANCE

We present the optimization performance of various acquisition functions (AFs) and their variants for each individual benchmark. These results form the basis for the performance ranking plots in Section 4. For synthetic benchmarks with known optimal values, we show the simple regret, whereas we show the best-observed function value at each iteration for the real-world benchmarks with unknown optima.

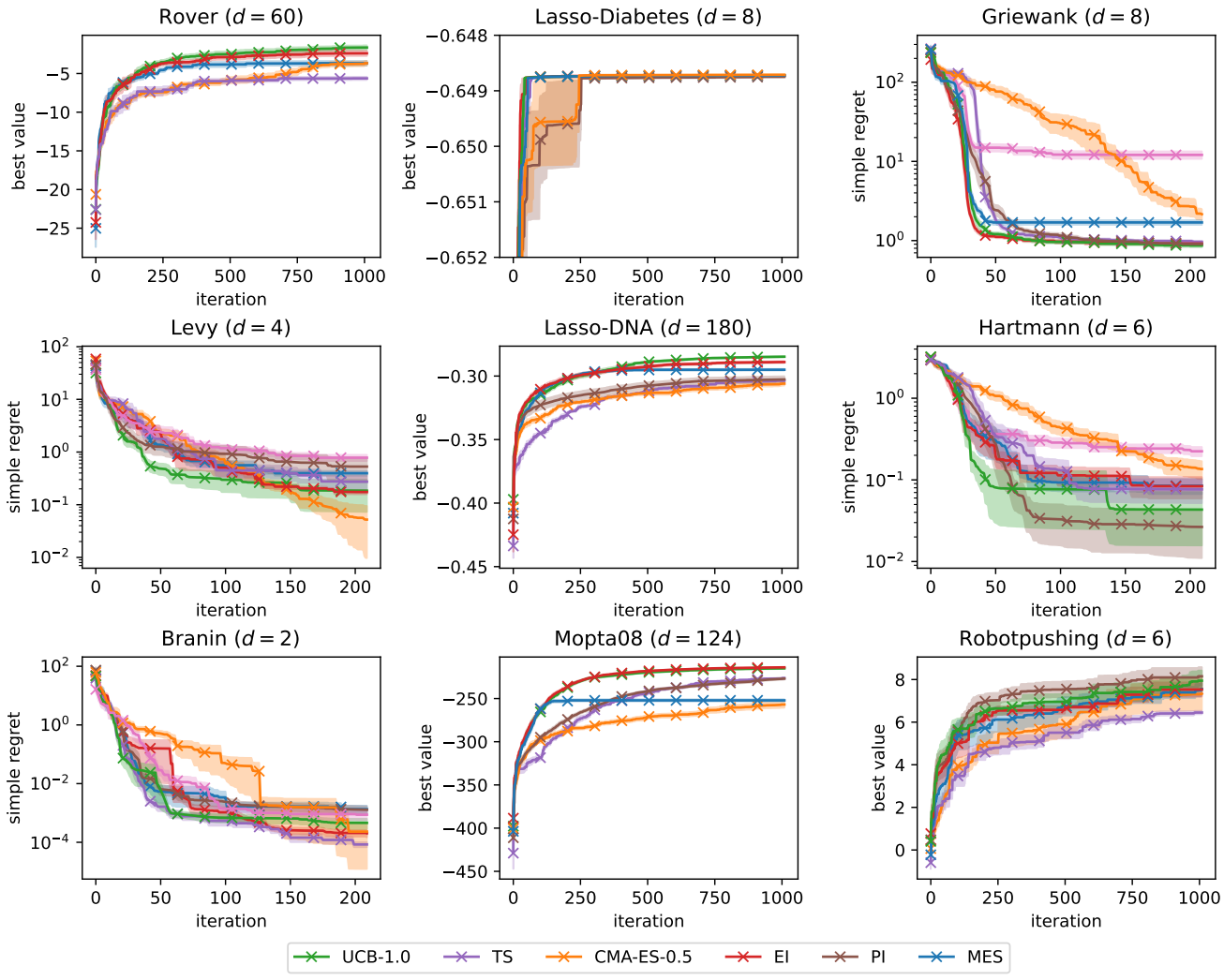


Figure 28: Optimization performance of the basic optimizer configuration on the different benchmarks.

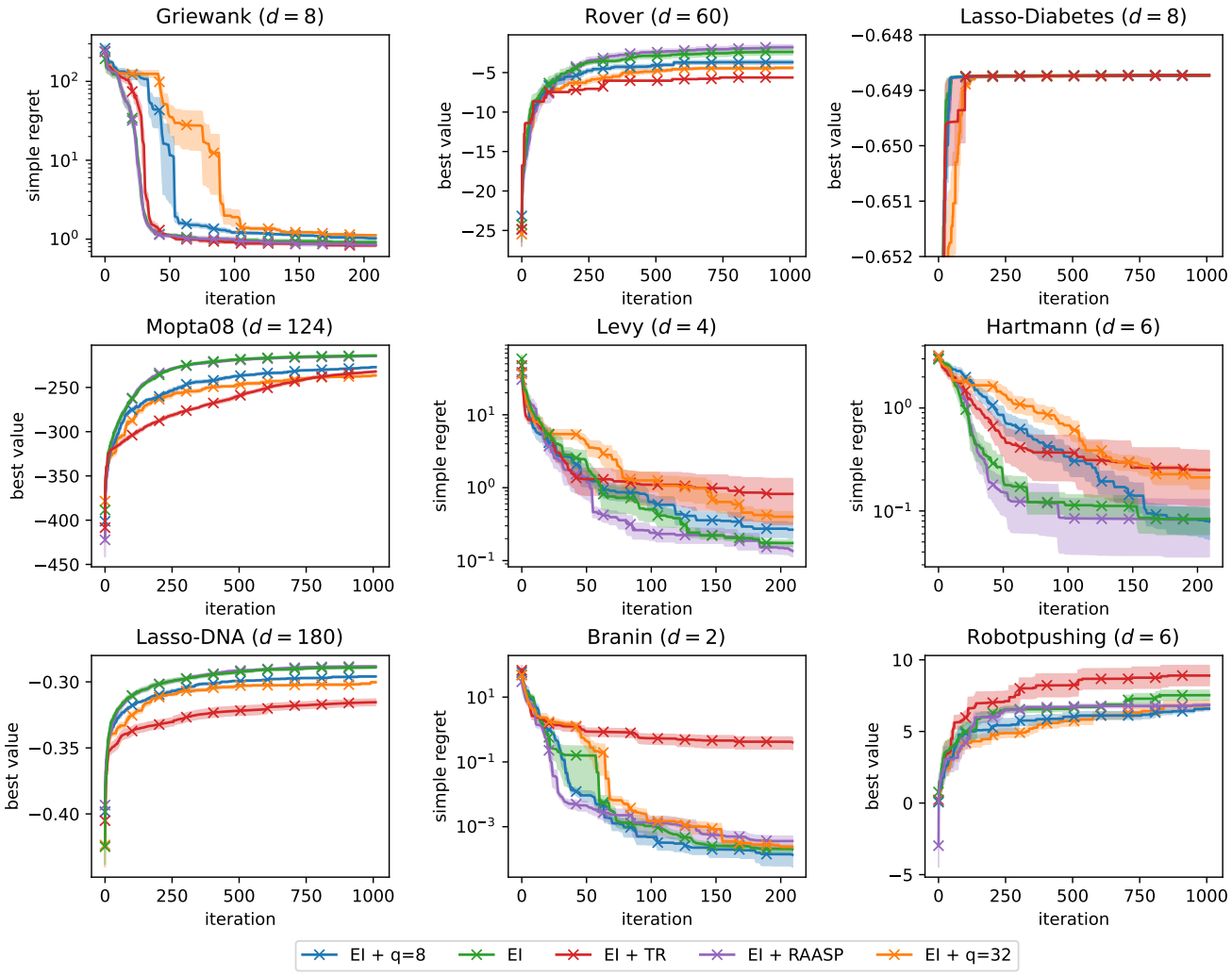


Figure 29: Optimization performance of the different EI variations on the benchmarks.

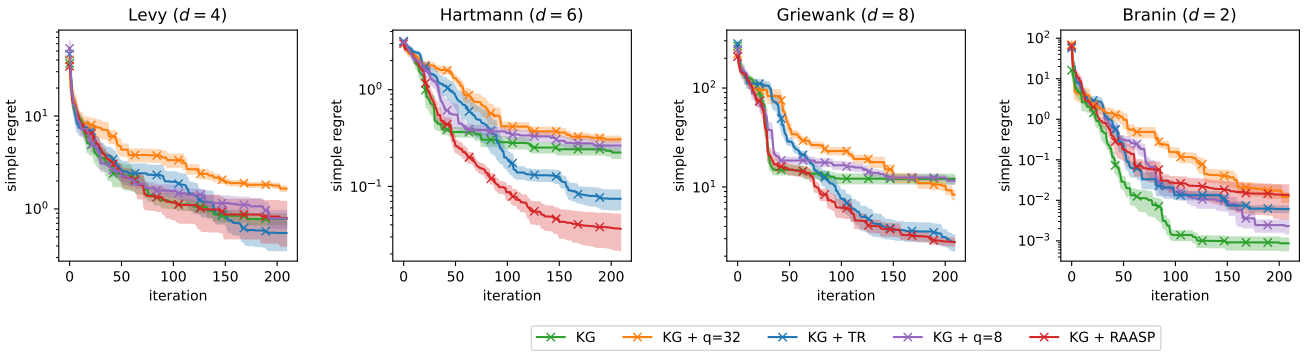


Figure 30: Optimization performance of the different KG variations on the benchmarks.

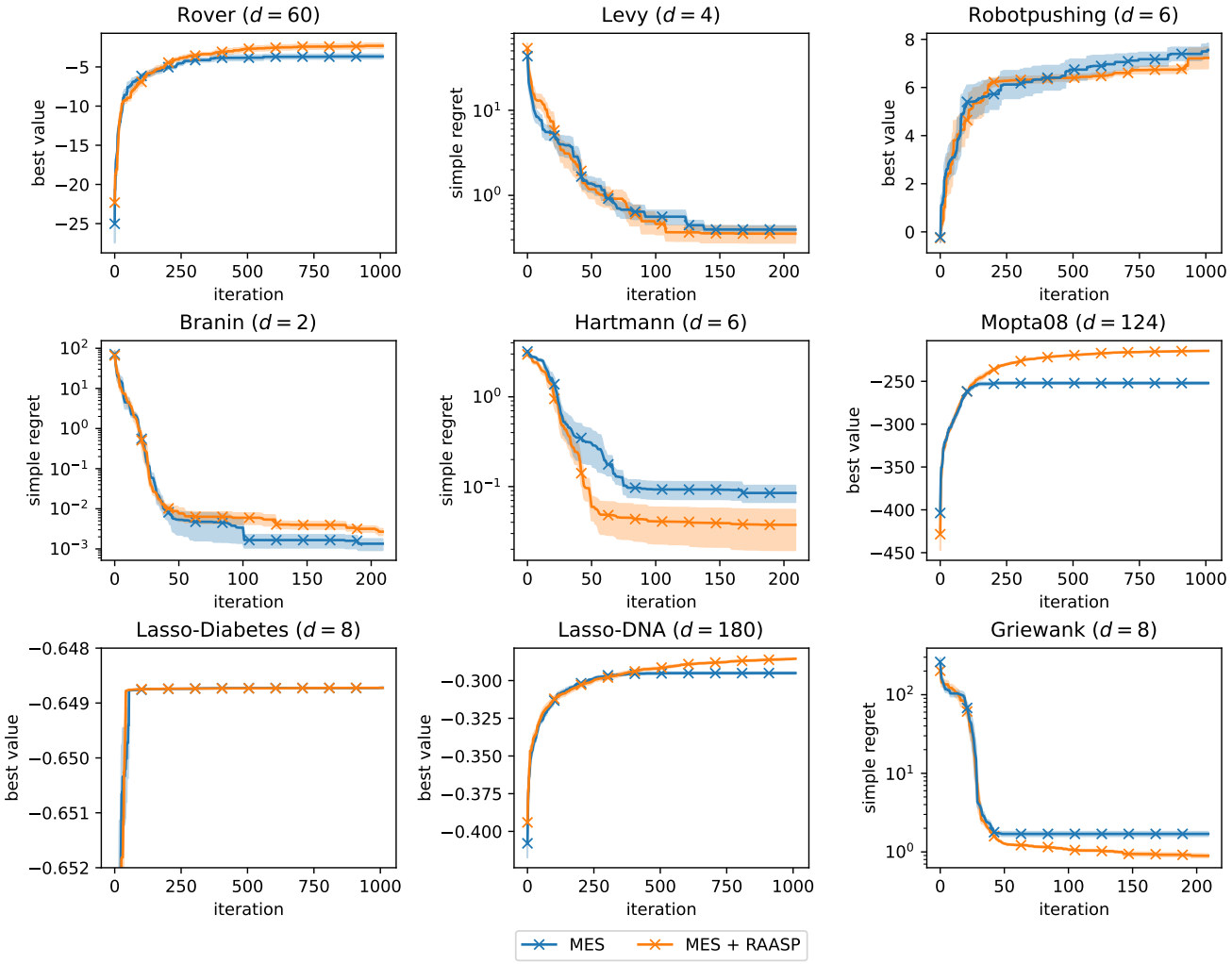


Figure 31: Optimization performance of the different MES variations on the benchmarks.

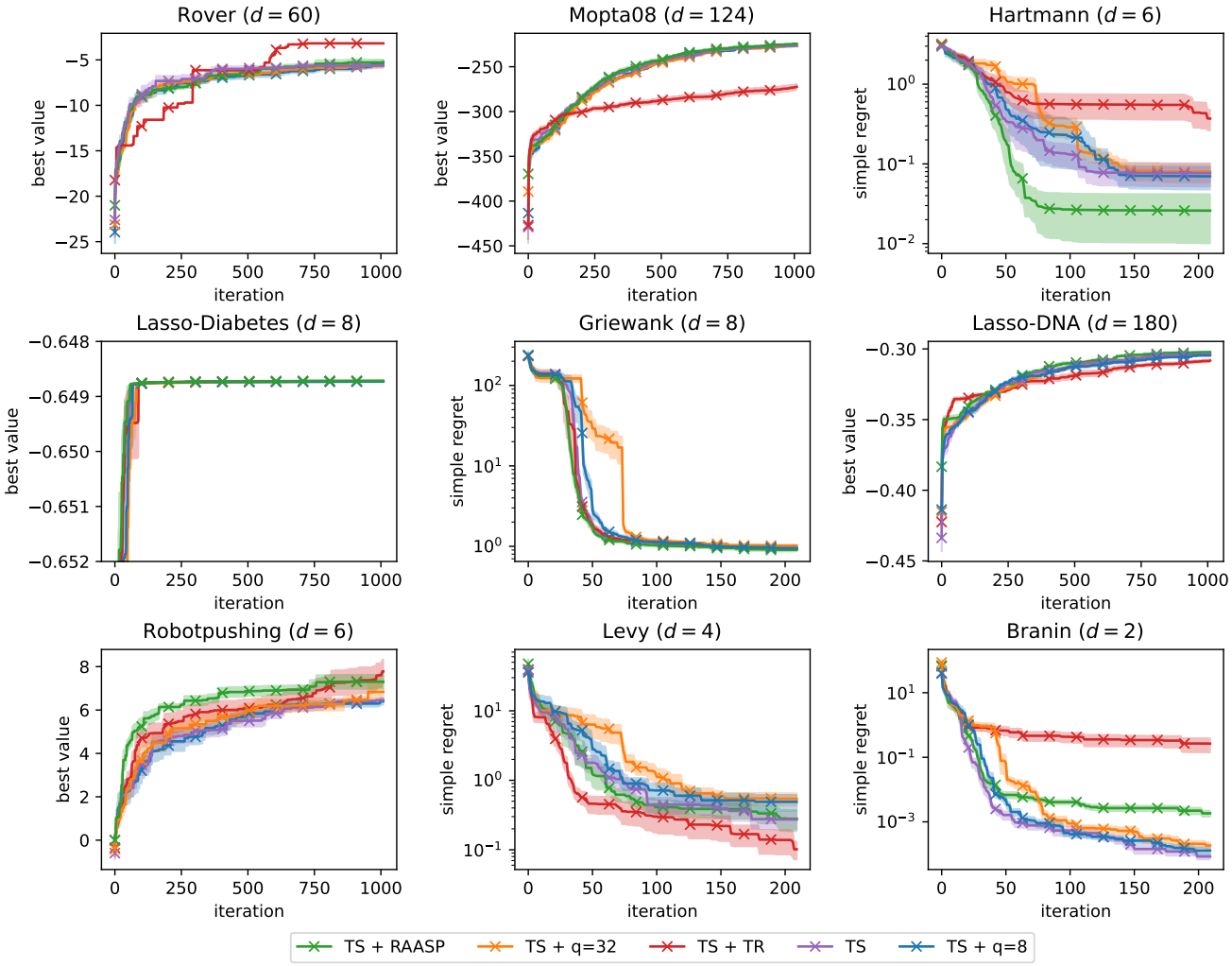


Figure 32: Optimization performance of the different TS variations on the benchmarks.

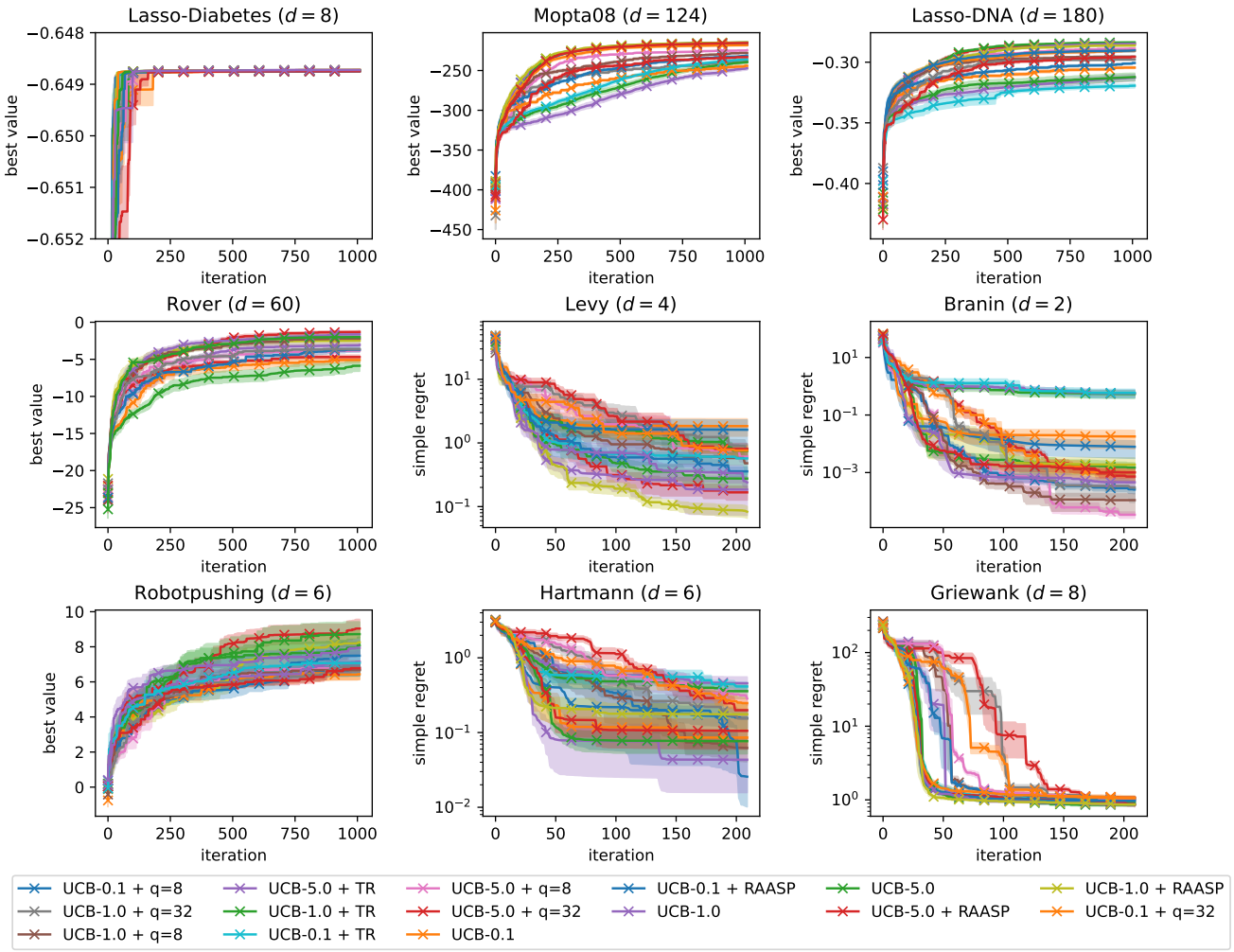


Figure 33: Optimization performance of the different UCB variations on the benchmarks.

E RUNTIME ANALYSIS

We analyze the runtimes for OTSD and OE. We measure OTSD and OE on random sequences of length 1000 in different dimensionalities, ranging from 10 – 1000 for OTSD and from 2 – 20 for OE. We repeat each experiment 20 times and observe the runtimes for calculating OTSD and OE. Figure 34 shows the runtimes for computing OTSD and OE. Computing OTSD is fast; for 1000 observation points in 10 dimensions it takes less than 200 ms and in 1000 dimensions less than 300 ms. OE is considerably more costly and restricted to low-dimensional spaces.

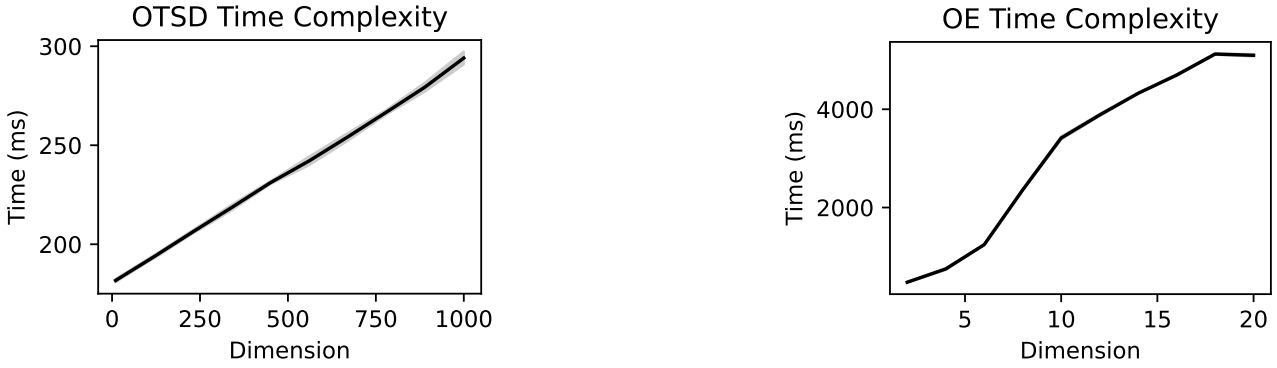


Figure 34: Runtimes for OTSD and OE in different dimensions. Empirically, OTSD scales linearly with the number of dimensions. OE is costlier.