Preference Exploration for Efficient
Bayesian Optimization with Multiple Outcomes

Zhiyuan Jerry Lin
Meta

Raul Astudillo
Cornell University

Peter I. Frazier
Cornell University

Eytan Bakshy
Meta

Abstract

We consider Bayesian optimization of expensive-to-evaluate experiments that generate vector-valued outcomes over which a decision-maker (DM) has preferences. These preferences are encoded by a utility function that is not known in closed form but can be estimated by asking the DM to express preferences over pairs of outcome vectors. To address this problem, we develop *Bayesian optimization with preference exploration*, a novel framework that alternates between interactive *real-time* preference learning with the DM via pairwise comparisons between outcomes, and Bayesian optimization with a learned compositional model of DM utility and outcomes. Within this framework, we propose preference exploration strategies specifically designed for this task, and demonstrate their performance via extensive simulation studies.

1 INTRODUCTION

Bayesian optimization (BO) is a sequential experimental design framework for efficient global optimization of black-box functions with expensive or time-consuming evaluations. It has succeeded in many real-world experimentation tasks, including materials design (Frazier and Wang, 2016; Packwood, 2017; Zhang et al., 2020), robot locomotion (Calandra et al., 2016), and internet experiments (Letham et al., 2019; Mao et al., 2019).

This work focuses on a common practical problem faced by decision-makers (DMs) who wish to apply BO to time-consuming experiments with multiple outcomes of interest. DMs have unknown preferences over outcomes which can be elucidated via a limited set of

interactions with the DM, and we wish to gather information through such interactions to support efficient experimentation. Such problems commonly arise in A/B testing (Bakshy et al., 2018) and simulation-based design (Maddox et al., 2021).

There are several possible approaches to BO with multiple outcomes in the literature, each with their own desiderata in our context. One approach is to have the DM express a fully-determined trade-off over outcomes via a function combining these outcomes into a single real-valued performance measure, and to perform single-objective BO with this function. Unfortunately, DMs are often unable to do this (Lepird et al., 2015).

A second approach is multi-objective BO (MOBO) (Hakanen and Knowles, 2017; Feliot et al., 2018; Abdolshah et al., 2019). MOBO aims to identify the entire feasible Pareto front but is typically inefficient because DMs are often interested only in a particular part of the Pareto front (Wang et al., 2017).

A third approach directly presents a DM sets of q designs (most commonly, pairs, i.e., $q = 2$) and asks the DM to express their preference over the sets. This data is used to model the DM's preferences over the designs directly (Brochu et al., 2008; Brochu, 2010; González et al., 2017; Siivola et al., 2020). We broadly refer to these methods as preferential BO (PBO).

PBO methods can be implemented in our context by performing time-consuming experiments for the designs in the sets selected by a PBO algorithm, and then presenting their outcomes to the DM for comparison. However, such an approach would be inefficient in terms of time, experiment resources, and DM attention since DMs must wait for experiment results to complete to further input their preferences, which can be disruptive and time-consuming. While higher throughput is possible if many designs are evaluated simultaneously (Siivola et al., 2020), the rate at which additional information can be gathered about the DM's preferences is limited by the time-consuming experiments. Instead, we might learn more with less DM time via queries generated in real time using existing

or hypothetical outcomes based on previously observed experimental data.

Toward overcoming the drawbacks of the above discussed approaches, we propose a novel human-in-the-loop algorithmic framework for optimizing multiple outcomes called *Bayesian optimization with preference exploration (BOPE)*. In this framework, outcomes arise from a time-consuming-to-evaluate function $f_{\text{true}} : \mathbb{R}^d \to \mathbb{R}^k$, and the DM's preferences can be viewed as originating from a utility function $g_{\text{true}} : \mathbb{R}^k \to \mathbb{R}$, which is unknown but can be learned through the DM's responses to *queries* in the form of comparisons between outcomes[1]. The goal is to solve

$$\max_{x \in \mathcal{X}} g_{\text{true}}(f_{\text{true}}(x)), \qquad (1)$$

where $\mathcal{X} \subset \mathbb{R}^d$ is the *design space*, using a limited number of queries to the DM and experiments (i.e., evaluations of f_{true}).

To do so, our framework iterates between two stages: *preference exploration* (PE) and *experimentation*. During a PE stage, an algorithm (a *PE strategy*) generates a query consisting of two outcome vectors (i.e., elements of $\mathbb{R}^k$) for the DM to compare. The DM states their preferred outcome, and another query is presented in real time based on the result. Outcome vectors in these queries need not be generated by evaluating f_{true}. However, as we will see later, leveraging the available knowledge about f_{true} can significantly improve performance. During an experimentation stage, an *experimentation strategy* chooses a batch of points in the design space at which f_{true} is evaluated. In a PE stage, previous DM queries and past experiment evaluations are used to choose the outcome vectors about which DM preference is elicited. Similarly, in a experimentation stage, all information gathered by this point is used to determine design points that will be evaluated.

This approach provides four benefits. First, relative to PBO, it supports models that decompose the latent objective function into separate models of the outcomes and the DM's preferences over these outcomes. This can improve prediction relative to PBO. Second, it supports greater flexibility than PBO when selecting user queries. Queries can be selected adaptively after each batch of experiments and can include hypothetical outcomes generated via a predictive model of f_{true}, or outcomes of designs evaluated at previous points in time. Third, relative to MOBO, learned preferences allow focusing experimental attention on the relevant portion of the Pareto front, reducing the number of experiments needed. Finally, in comparison with MOBO, our approach automatically handles non-monotone pref-
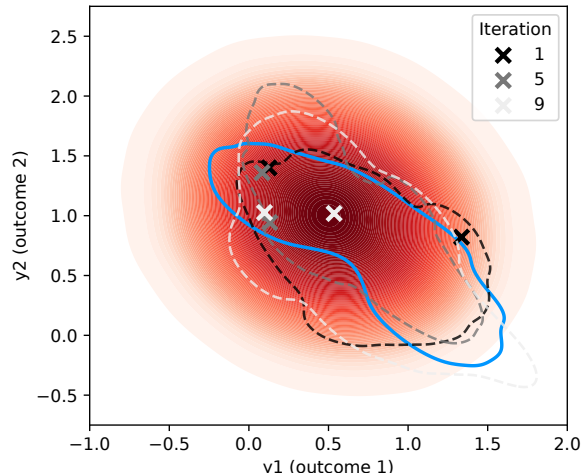

Figure 1: Preference exploration for a problem with a two-dimensional outcome space. The heatmap illustrates the DM's utility function $g_{\text{true}}(y)$ and the blue line circumscribes the space of possible outcomes achievable by $f_{\text{true}}(x)$ for any design $x \in \mathcal{X}$. Evaluation of $f_{\text{true}}(\cdot)$ is time-consuming. Our framework aims to collect preference data to support Bayesian optimization of $g_{\text{true}}(f_{\text{true}}(\cdot))$ with the aid of probabilistic surrogate models of outcomes (f) and DM utilities (g). $\times$s show outcome vectors presented to the DM by EUBO-$\tilde{f}$, over multiple iterations. Each iteration queries regions likely to be of highest utility to the DM according to g within search sets defined by independent sample paths $\tilde{f}$ from f (dashed loops). This procedure helps learn a g that may be used to select high-utility experiments.

erences over outcomes. This might arise, e.g., when designing a material that should neither be too stiff nor too flexible, or in drug discovery when a chemical's concentration in the blood should fall near a target.

This new workflow includes a key challenge unaddressed by prior work: how should preference information be gathered to best support optimization in such real-world contexts? Responding to this question, we examine several PE strategies and show that the most successful ones leverage information about the posterior distribution of outcomes achievable under f_{true}. Finally, we develop a one-step Bayes optimal PE strategy for collecting preference information to solve (1).

Figure 1 illustrates one of our proposed PE strategies, EUBO-$\tilde{f}$. Rather than performing preference learning across all possible outcome vectors, which would take many queries, EUBO-$\tilde{f}$ focuses on queries comparing outcomes that are likely to be achievable. This guides experiments toward regions of high utility to the DM in a small number of queries.

---

[1]For simplicity we consider pairwise comparisons, but generalizing BOPE to other comparisons is straightforward.

Contributions Our contributions are as follows:

- We propose Bayesian optimization with preference exploration (BOPE), a novel human-in-the-loop framework for BO of time-consuming experiments that generate vector-valued outcomes over which a DM has unknown preferences. This framework reduces experimental cost and DM time over existing MOBO and PBO approaches.

- We develop a principled PE strategy for adaptively selecting queries to present to the DM. The DM's answers to these queries localize the parts of the achievable region of outcomes that are of importance to the DM, and enable selection of high-utility designs to evaluate via experimentation.

- We evaluate our approach on synthetic and real-world simulation problems, such as multi-objective vehicle design, demonstrating that PE-based approaches significantly outperform MOBO, PBO, and other natural baselines.

2 OTHER RELATED WORK

A related stream of research has developed techniques for efficient elicitation of the DM's preferences over outcomes that are known *a priori* without the need for expensive evaluation (Chajewska et al., 2000; Boutilier, 2002; Fürnkranz and Hüllermeier, 2003; Chu and Ghahramani, 2005; Viappiani and Boutilier, 2010; Dewancker et al., 2016). While we leverage ideas from these works, they are not directly applicable in our setting, where outcomes are unknown and expensive-to-evaluate. Additionally, most of this literature assumes that there are a finite number of outcome vectors, while our space of potential outcomes is uncountably infinite.

Our work builds on Astudillo and Frazier (2020) and Lin et al. (2020). The former considers sequential BO of vector-valued functions on behalf of a DM with unknown preferences using the same decomposition of outcomes and utilities considered here and proposes a strategy for the experimentation stage only. The present paper builds off of a previous workshop paper by Lin et al. (2020) which considers a similar problem setup as Astudillo and Frazier (2020) but focuses on batch-based PE and experimentation to support the types of workflows found in industry A/B testing settings. The present paper formalizes this setup, and provides a principled solution to the problem.

Finally, due to the composite structure of problem (1), our work is also related to BO of composite objective functions (Astudillo and Frazier, 2019, 2021a,b), and we leverage similar computational techniques.

3 BO WITH PREFERENCE EXPLORATION

Here, we describe our framework, including the workflow and core models, and introduce key considerations for the design of strategies for PE and experimentation.

3.1 Workflow

In our framework, the goal is to find a solution to (1) by learning the functions g_{true} and f_{true} while alternating between two stages: *preference exploration* and *experimentation.*

A PE stage is a short uninterrupted period of time in which the DM interactively expresses preferences over multiple pairs of outcome vectors that does not involve the collection of new values from f_{true}. We refer to the pair of outcome vectors presented to the DM as a *query* and to the DM's answer as the *response.*

An experimentation stage is a period of time in which one or more evaluations of the outcome function f_{true} are performed, typically in parallel. Our presentation assumes that PE and experimentation stages alternate. As a special case, this can include situations where PE is performed only once after an initial round of experimentation, so as to take the human out of the loop for subsequent experiments. For example, in the context of internet experimentation, an experimentation stage would entail running a batch of A/B tests in parallel and a PE stage could be an interactive session with a data scientist where pairs of (potentially hypothetical) experimental outcomes are compared.

Once all PE and experimentation stages are complete, the DM is shown the outcome vectors for all experiments that have been performed, and the DM selects their preferred design.[2]

3.2 Outcome and Preference Models

We utilize two probabilistic surrogate models, one for the outcome function, f_{true}, and one for the utility function, g_{true}.

To model the outcome function, f_{true}, we use a multi-output Gaussian process (GP), f, characterized by a prior mean function $\mu_0^f : \mathcal{X} \to \mathbb{R}^k$, and a prior covariance function, $K_0^f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{k \times k}$. Given a dataset of n (potentially noisy) observations of the outcome function, $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$, the probabilistic surrogate model of f_{true} is then given by the posterior distribution of f given $\mathcal{D}_n$, which is again a multi-output GP

---

[2]In practice, the learned utility may be used to rank these items in advance to reduce the cognitive burden of sorting through all evaluated designs.

with mean and covariance functions $\mu_n^f : \mathcal{X} \to \mathbb{R}^k$ and $K_n^f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{k \times k}$ that can be computed in closed form in terms of μ_0^f and K_0^f.

The utility function, g_{true}, is also modeled using a GP, g, which again requires specifying a prior mean function, $\mu_0^g : \mathbb{R}^k \to \mathbb{R}$, and a prior covariance function, $K_0^g : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$.

Given a query (y_1, y_2) constituted by two outcome vectors, we let $r(y_1, y_2) \in \{1, 2\}$ indicate whether the DM preferred the first or second outcome vector offered. Following Chu and Ghahramani (2005), we assume that the DM's responses are distributed according to a probit likelihood of the form

$$\mathbb{P}(r(y_1, y_2) = 1 \mid g(y_1), g(y_2)) = \Phi\left(\frac{g(y_1) - g(y_2)}{\sqrt{2}\lambda}\right),$$

where λ is a hyperparameter that can be estimated along with the other hyperparameters of the model, and Φ is the standard normal CDF.

In our experiments, we use the Laplace approximation suggested by Chu and Ghahramani (2005), which results in an approximate posterior of g that is again a GP. When we have observed the results of m queries, $\mathcal{P}_m = \{(y_{1,j}, y_{2,j}, r(y_{1,j}, y_{2,j}))\}_{j=1}^m$, we let μ_m^g and K_m^g refer to the mean and covariance functions of this approximate GP posterior.

3.3 Preference Exploration Strategies

Preference exploration strategies aim to select queries $(y_1, y_2) \in \mathcal{Y} \times \mathcal{Y}$, where $\mathcal{Y} \subseteq \mathbb{R}^k$, so as to best support experiment selection. Here, we introduce three classes of PE strategies investigated in this work.

PE Strategies That Learn Preferences Over a Prior Region of Interest Our first class of PE strategies requires choosing a prior set $\mathcal{Y}$ likely to contain most or all of the achievable region $f_{\text{true}}(\mathcal{X}) = \{f_{\text{true}}(x) : x \in \mathcal{X}\}$, so that preferences over $\mathcal{Y}$ are highly informative of preferences over $f_{\text{true}}(\mathcal{X})$. It then chooses queries to learn DM preferences over $\mathcal{Y}$. In the simplest case, $\mathcal{Y}$ can be a hyperrectangle bounding a likely minimum and maximum for each outcome provided by the DM. Alternatively, $\mathcal{Y}$ could be estimated via a meta-analysis of related experiments, and could also incorporate information about how outcomes covary across experiments.

There is a tradeoff in choosing the bounds of $\mathcal{Y}$. Choosing $\mathcal{Y}$ to be too small risks excluding relevant potential outcomes. Choosing $\mathcal{Y}$ to be much larger than $f_{\text{true}}(\mathcal{X})$ can cause over-exploration of areas not relevant to the optimization task. This may occur with hyper-rectangular $\mathcal{Y}$ if correlated outcomes make $f_{\text{true}}(\mathcal{X})$ much smaller than $\mathcal{Y}$.

Two policies for generating queries to learn preferences over $\mathcal{Y}$ are: selecting queries uniformly at random over $\mathcal{Y} \times \mathcal{Y}$; and selecting queries by maximizing an active learning acquisition function (AF) such as Bayesian active learning by disagreement (BALD) (Houlsby et al., 2011) over $\mathcal{Y} \times \mathcal{Y}$.

PE Strategies That Leverage Direct Experimental Data Another class of PE strategies samples many plausible achievable regions based on data from f_{true}. For each such region $\mathcal{Y}$, it seeks to learn the DM's preferences over $\mathcal{Y}$. Learning these preferences allows us to eventually learn the best point in the true achievable region, despite not knowing this region.

Probabilistic surrogate models provide a natural mechanism for sampling plausible achievable regions. In many PE strategies considered here, for each preference query, a GP sample path $\tilde{f}$ is first drawn from f, implying an associated achievable region $\mathcal{Y} = \tilde{f}(\mathcal{X})$. Then a query is designed to improve our knowledge of the DM's most preferred design in $\mathcal{Y}$. Compared to the former class of PE strategies that learn preferences over a prior region of interest, this approach aims to reduce the number of queries needed by learning a separate ranking over many smaller sets $\mathcal{Y}$. As we learn more about the achievable region through experimentation, the sequence of sampled sets $\mathcal{Y}$ concentrates, further reducing the size of the query space.

We consider three strategies for learning preferences over $\mathcal{Y} = \tilde{f}(\mathcal{X})$: random search, BALD, and a novel AF called EUBO which is introduced in §4.1 and aims to find the best query over a known set of outcomes. §4.3 also derives an approximation of a one-step optimal AF for BOPE that has a similar structure but uses a different choice of $\mathcal{Y}$.

Optimization of PE AFs Several of the PE strategies discussed above require optimizing an AF α over $\mathcal{Y} \times \mathcal{Y}$. In several cases, $\mathcal{Y}$ can be written as $\mathcal{Y} = \{h(x) : x \in \mathcal{X}\}$ for some deterministic function h. This makes optimization convenient, since is allows for optimization over a known domain, $\mathcal{X}$: $\text{argmax}_{y_1, y_2 \in \mathcal{Y}} \alpha(y_1, y_2) = \text{argmax}_{x_1, x_2 \in \mathcal{X}} \alpha(h(x_1), h(x_2))$. Thus, to find the maximizer of $\alpha(y_1, y_2)$ over $y_1, y_2 \in \mathcal{Y}$, it is sufficient to find the maximizer of $\alpha(h(x_1), h(x_2))$ over $x_1, x_2 \in \mathcal{X}$.

3.4 Experiment Selection Strategies

Here, we discuss the strategy we use to select the designs at which f_{true} is evaluated during the experimentation stages. Since the focus of our work is on PE strategies, we restrict our attention to a single experiment selection strategy. We propose a generalization of the expected improvement under utility uncertainty (EIUU) AF, introduced by Astudillo and Frazier (2020),

to support noisy and parallel evaluations with a non-parametric utility model g. This AF integrates over the uncertainty of both f and g when selecting the design points.

Formally, for a batch of q points $x_{1:q} = (x_1, \ldots, x_q) \in \mathcal{X}^q$, this AF is defined by

$$\text{qNEIUU}(x_{1:q}) =$$
$$\mathbb{E}_{m,n}\left[\{\max g(f(x_{1:q})) - \max g(f(X_n))\}^+\right],$$

where $\mathbb{E}_{m,n}[\cdot] = \mathbb{E}[\cdot \mid \mathcal{P}_m, \mathcal{D}_n]$ denotes the conditional expectation given the data from m queries and n experiments, $\{\cdot\}^+$ denotes the positive part function, and, making a slight abuse of notation, we define $\max g(f(x_{1:q})) = \max_{i=1,\ldots,q} g(f(x_i))$ and $\max g(f(X_n)) = \max_{(x,y) \in \mathcal{D}_n} g(f(x))$.

qNEIUU can be straightforwardly implemented as a Monte Carlo AF by applying the reparametrization trick to both f and g, and optimized via sample average approximation (SAA) (Wilson et al., 2018; Balandat et al., 2020). We refer the reader to §A.1 in the supplementary material (SM) for more details on the implementation of qNEIUU.

4 ONE-STEP OPTIMAL PREFERENCE EXPLORATION

Here we present a principled PE strategy derived using a one-step Bayes optimality analysis. We begin by describing a one-step optimal strategy for learning preferences over a known set of outcome vectors as well as results regarding its efficient computation. We then propose a one-step optimal PE strategy that formally accounts for uncertainty over the set of achievable outcomes. This strategy is not practical for real-time learning, but we provide a faster principled approximation using insights developed in the case with known achievable outcomes.

4.1 Preference Exploration Over a Known Set of Outcomes

Here we assume that the space of achievable outcomes is known and denote it by $\mathcal{Y}$.

To motivate our AF, we ask the following rhetorical question: If we had to offer a single outcome vector $y^* \in \mathcal{Y}$ to the DM using the available information so far, what would the right choice be? We argue that a sensible choice is y^* so that the expected utility received by the DM is maximal; i.e., $y^* \in \operatorname{argmax}_{y \in \mathcal{Y}} \mathbb{E}_m[g(y)]$, where $\mathbb{E}_m$ denotes the conditional expectation given $\mathcal{P}_m$ (i.e., $\mathbb{E}_m[\cdot] = \mathbb{E}[\cdot \mid \mathcal{P}_m]$). Following this logic, if we were allowed to ask an additional query (y_1, y_2) and observe the DM's response $r(y_1, y_2)$, y^* would now

be chosen so that $y^* \in \operatorname{argmax}_{y \in \mathcal{Y}} \mathbb{E}_{m+1}[g(y)]$, where $\mathbb{E}_{m+1}[\cdot] = \mathbb{E}[\cdot \mid \mathcal{P}_m \cup \{(y_1, y_2, r(y_1, y_2))\}]$. Thus,

$$\max_{y \in \mathcal{Y}} \mathbb{E}_{m+1}[g(y)] - \max_{y \in \mathcal{Y}} \mathbb{E}_m[g(y)]$$

quantifies the difference in (expected) utility obtained by the DM due to the additional query. Our AF can now be defined as the expectation of this difference given the information available so far; i.e.,

$$V(y_1, y_2) = \mathbb{E}_m\left[\max_{y \in \mathcal{Y}} \mathbb{E}_{m+1}[g(y)] - \max_{y \in \mathcal{Y}} \mathbb{E}_m[g(y)]\right],$$

where the dependence of the right-hand-side on (y_1, y_2) is made implicit by our notation $\mathbb{E}_{m+1}[\cdot] = \mathbb{E}[\cdot \mid \mathcal{P}_m \cup \{(y_1, y_2, r(y_1, y_2))\}]$. We also note that the term $\max_{y \in \mathcal{Y}} \mathbb{E}_m[g(y)]$ does not depend on (y_1, y_2) and thus can be disregarded when maximizing V. This AF can be considered to be in the knowledge gradient family of AFs (Frazier, 2018), because it values information according to its impact on the maximum posterior expected utility.

4.1.1 Expected Utility of the Best Outcome

The acquisition function V defined above is challenging to maximize directly due to its nested structure, as is typical of knowledge gradient AFs. Fortunately, the following theorem shows that maximizing V is equivalent to maximizing another AF that is easier to optimize.

We define the *expected utility of the best option (EUBO)*,

$$\text{EUBO}(y_1, y_2) = \mathbb{E}_m\left[\max\{g(y_1), g(y_2)\}\right],$$

where the expectation is over the posterior of the DM utility g at the time the query is chosen. V and EUBO are related via the following result.

Theorem 1. *If $\lambda = 0$, and the posterior mean μ_m^g and posterior covariance K_m^g are both continuous, then*

$$\operatorname*{argmax}_{y_1, y_2 \in \mathcal{Y}} \text{EUBO}(y_1, y_2) \subseteq \operatorname*{argmax}_{y_1, y_2 \in \mathbb{R}^k} V(y_1, y_2)$$

and the left-hand side is non-empty.

Thus, the above result shows that, when DM responses occur without error, one can find a maximizer of V by maximizing EUBO instead, which, as we argue below, is a significantly simpler task. While this holds for $\lambda = 0$ only, the following result shows that maximizing EUBO yields a high-quality solution even if $\lambda > 0$.

Theorem 2. *Denote V as V_λ to make the dependence on λ explicit, and let $(y_1^*, y_2^*) \in \operatorname{argmax}_{y_1, y_2} \text{EUBO}(y_1, y_2)$. Then,*

$$V_\lambda(y_1^*, y_2^*) \geq \max_{y_1, y_2 \in \mathcal{Y}} V_0(y_1, y_2) - \lambda C,$$

where $C = e^{-1/2}/\sqrt{2}$.

Theorem 2 provides a lower bound on the acquisition value of a maximizer of EUBO evaluated at V in the presence of comparison noise. The proofs of Theorems 1 and 2 can be found in the SM §B.

Maximizing EUBO is easier than V for two reasons. First, computing EUBO does not require solving the inner optimization problem $\max_{y \in \mathcal{Y}} \mathbb{E}_{m+1}[g(y)]$ required by V. Second, EUBO can be expressed in closed form under the approximate Gaussian posterior of g implied by the Laplace approximation, as shown in SM §B.2.1.

4.2 Preference Exploration Under an Unknown Set of Achievable Outcomes

We now describe a PE strategy that formally takes into account uncertainty on the set of achievable outcomes. To support our analysis, here we assume that evaluations of the outcome function are noise-free.

As in the previous subsection, we derive a PE strategy using a one-step optimality analysis. Formally, this strategy selects the query (y_1, y_2) that is optimal with respect to the following sequence of actions (which constitute one step):

1. Select query $(y_1, y_2) \in \mathbb{R}^k \times \mathbb{R}^k$ and observe DM's response $r(y_1, y_2)$,

2. Select design $x \in \mathcal{X}$ and observe outcome $f(x)$,

3. Obtain reward $\max_{i=1,\ldots,n+1} \mathbb{E}_{m+1,n+1}[g(f(x_i))]$, where we define $x_{n+1} = x$ and $\mathbb{E}_{m+1,n+1}[\cdot]$ denotes $\mathbb{E}[\cdot \mid \mathcal{P}_m \cup \{(y_1, y_2, r(y_1, y_2))\}, \mathcal{D}_n \cup \{(x, f(x))\}]$.

The optimal query can be found by solving $\max_{y_1, y_2 \in \mathbb{R}^k} W(y_1, y_2)$, where

$$W(y_1, y_2) = \mathbb{E}_{m,n}\Big[\max_{x_{n+1} \in \mathcal{X}} \mathbb{E}_{m+1,n}\big[\max_{i=1,\ldots,n+1} \mathbb{E}_{m+1,n+1}[g(f(x_i))]\big]\Big]$$

and $\mathbb{E}_{m+1,n}[\cdot] = \mathbb{E}[\cdot \mid \mathcal{P}_m \cup \{(y_1, y_2, r(y_1, y_2))\}, \mathcal{D}_n]$

This strategy is similar in spirit to the one described in the previous subsection. However, there are two key differences, which originate due to the set of achievable outcomes being unknown. First, there is an additional action between query selection and reward collection. Here, one additional evaluation of the outcome function is performed, thus enforcing the need to gather preference information to support experimentation (i.e., evaluations of the outcome function). Second, since the set of achievable outcomes is unknown, the reward is computed over the outcomes observed so far only.

4.3 Efficient Approximate Maximization of W Via a Single-Sample Approximation

Unsurprisingly, W is quite hard to compute and optimize due to its nested structure. In principle, one could aim to adapt optimization strategies for lookahead AFs (see, e.g., Balandat et al. 2020; Jiang et al. 2020). However, these methods are quite computationally expensive, with run times on the order of several minutes per acquisition, making them impractical in the context of real-time PE. Instead, we derive an efficient approximate optimization scheme based on a single-sample approximation of W with respect to the uncertainty on $f(x_{n+1})$.

Leveraging the approximate GP distribution over g induced by the Laplace approximation, we write $\mathbb{E}_{m+1,n+1}[g(f(x_i))]$ as $\mu^g_{m+1}(f(x_i))$ for $i = 1, \ldots, n+1$,[3] and define $\mu^*_{m+1,n} = \max_{i=1,\ldots,n} \mu^g_{m+1}(f(x_i))$. Applying the reparametrization trick on $f(x_{n+1})$, W can be rewritten as

$$W(y_1, y_2) = \mathbb{E}_{m,n}[\max_{x \in \mathcal{X}} \mathbb{E}_{m+1,n}[\max\{\mu^*_{m+1,n}, \mu^g_{m+1}(\zeta_n(x; Z))\}]],$$

where $\zeta_n(x; Z) = \mu^f_n(x) + C^f_n(x)Z$, $C^f_n(x)$ is the lower Cholesky factor of $K^f_n(x, x)$, and the (conditional) distribution of Z is the standard normal. In the expression above, the inner expectation is over Z and the outer expectation is over $r(y_1, y_2)$.

If we approximate the expression above using a single sample from Z, which we denote by $\tilde{Z}$, and making a slight abuse of notation, we obtain the approximation $W(y_1, y_2) \approx W(y_1, y_2; \tilde{Z})$, where

$$W(y_1, y_2; \tilde{Z}) = \mathbb{E}_{m,n}[\max_{x \in \mathcal{X}} \max\{\mu^*_{m+1,n}, \mu^g_{m+1}(\zeta_n(x; \tilde{Z}))\}],$$

and $\tilde{Z}$ is deterministic in the expectation above. Moreover, since $\zeta_n(x_i; \tilde{Z}) = f(x_i)$ for previously evaluated points $i = 1, \ldots, n$, $\max_{x \in \mathcal{X}} \mu^g_{m+1}(\zeta_n(x; \tilde{Z})) \geq \mu^*_{m+1,n}$ and thus $\max_{x \in \mathcal{X}} \max\{\mu^*_{m+1,n}, \mu^g_{m+1}(\zeta_n(x; \tilde{Z}))\}$ can be simplified to $\max_{x \in \mathcal{X}} \mu^g_{m+1}(\zeta_n(x; \tilde{Z}))$. Thus,

$$W(y_1, y_2; \tilde{Z}) = \mathbb{E}_{m,n}[\max_{x \in \mathcal{X}} \mu^g_{m+1}(\zeta_n(x; \tilde{Z}))].$$

We can now use the machinery developed in §4.1 to efficiently maximize $W(y_1, y_2; \tilde{Z})$. Concretely, if we let $\mathcal{Y} = \{\zeta_n(x; \tilde{Z}) : x \in \mathcal{X}\}$, it follows from Theorem 1 that $\text{argmax}_{y_1, y_2 \in \mathcal{Y}} \text{EUBO}(y_1, y_2) \subseteq \text{argmax}_{y_1, y_2 \in \mathcal{Y}} W(y_1, y_2; \tilde{Z})$ when $\lambda = 0$. Analogously,

---

[3]We leverage this closed form expression to simplify our notation but this is not critical. Our analysis holds even if $\mathbb{E}_{m+1,n+1}[g(f(x_i))]$ does not have a closed form expression.

Theorem 2 provides a guarantee on the quality of a query in $\text{argmax}_{y_1, y_2 \in \mathcal{Y}} \text{EUBO}(y_1, y_2)$ when $\lambda > 0$. We call the resulting PE strategy EUBO-ζ.

While EUBO-ζ is derived as an approximation of a one-step optimal strategy, it has a structure similar to that of the strategies discussed in §3.3: it builds a set $\mathcal{Y}$ using f and then selects a query in $\mathcal{Y} \times \mathcal{Y}$ by maximizing an AF. This perspective leads us to consider the PE strategy that chooses queries by maximizing $\text{EUBO}(y_1, y_2)$ over $y_1, y_2 \in \tilde{f}(\mathcal{X})$. We call the resulting strategy EUBO-$\tilde{f}$. This variation of EUBO provides a more intuitive interpretation than EUBO-ζ.

5 NUMERICAL EXPERIMENTS

We evaluate our proposed strategies on real-world and synthetic outcome functions as well as several utility functions. The main text considers four test problems: vehicle safety ($d = 5, k = 3$) (Liao et al., 2008; Tanabe and Ishibuchi, 2020), DTLZ2 (Deb et al., 2005) ($d = 4, k = 8$), OSY ($d = 6, k = 8$) (Deb and Jain, 2013), and car cab design ($d = 7, k = 9$) problems (Tanabe and Ishibuchi, 2020). These test problems are matched with several utility functions: linear and piecewise linear utilities, the product of Kumaraswamy distribution CDFs, modeling soft constraints, and the L1 distance from a Pareto-optimal point. All test problems are described in detail in §E of the SM. Results are qualitatively similar to those presented here. Table 3 in the SM summarizes all test outcome and utility functions considered in this work.

With these outcome and utility functions, we perform three types of simulation-based experiments. §5.1 considers a single PE stage. With data from a single batch of experiments, we train a surrogate outcome model f, then use PE strategies to identify high utility designs in a single PE stage. §5.2 considers the case in which multiple PE and experimentation stages are interleaved, and evaluates the maximum utility found over several rounds of BOPE for each PE strategy. Finally, §5.3 considers a setting in which all preference exploration occurs in a single stage, and this is followed by multiple batches of experimentation without further intervention from the DM. All simulations compare EUBO-based PE strategies against several other PE strategies. Unless otherwise noted, we use the qNEIUU experiment selection strategy.

To emulate noise in preferences expressed by human DMs, simulated DMs in all experiments select the option with lower utility 10% of the time. In §D.5 of the SM, we also experiment with a different probit comparison error and observe similar results.

Complete simulation results with further baselines, test problems, and settings are available in §D of the SM.

Acquisition strategies for PE. We examine several PE strategies described in §3.3. This includes EUBO-ζ and methods selecting queries from one of two sets $\mathcal{Y}$: (i) a hyper-rectangle $\mathcal{Y}_0$ bounding $f_{\text{true}}(\mathcal{X})$, and (ii) $\tilde{f}(\mathcal{X})$. $\mathcal{Y}_0$ is constructed to be the smallest hyper-rectangle that contains $f_{\text{true}}(\mathcal{X})$, estimated via 10^8 Monte Carlo samples from $\mathcal{X}$. Thus, $\mathcal{Y}_0$ provides optimistic baseline of a DM that can perfectly characterize the upper and lower (box) bounds of f_{true}. $\tilde{f}$ is sampled from the posterior on f via random Fourier features with 512 basis functions (Rahimi and Recht, 2007).

To select queries from these two sets $\mathcal{Y}$, we consider random search, BALD, and EUBO. We name these algorithms via their sampling strategy and choice of $\mathcal{Y}$: Random-$\mathcal{Y}_0$ and Random-$\tilde{f}$, and similarly for EUBO and BALD. BALD-$\mathcal{Y}$ (for both choices of $\mathcal{Y}$) is estimated via quasi-Monte Carlo (QMC), and selects designs $x \in \mathcal{X}$ to reduce the posterior uncertainty of g over $\mathcal{Y}$. Variants of EUBO are computed using the closed form expression derived in §B.2.1 of the SM.

All algorithms are implemented in BoTorch (Balandat et al., 2020). All AFs are optimized via SAA using L-BFGS-B. We use a Matérn 5/2 ARD covariance function for the outcome model and RBF ARD kernel for the preference model. We refer the reader to §C of the SM for additional implementation details.

5.1 Identifying High Utility Designs with PE

We first examine how the proposed PE strategies identify design points whose outcomes have high utilities during a single PE stage. We first evaluate the outcome function at a batch of quasi-random design points and fit a multi-output GP to the observed data. Outcome functions with five or fewer input dimensions receive 16 initial designs and the remainder receive 32. We then initialize the preference model using pairs of random designs from the initial batch for the first $2k$ pairwise comparisons, followed by comparisons acquired via PE strategies. The outcome surrogate model remains unchanged.

We plot the true utility of our best guess at the utility-optimal design after every 5 pairwise comparisons. To do so, we maximize $\mathbb{E}[g(f(x))]$ over $x \in \mathcal{X}$ where the expectation is taken under the posterior given all available query responses and experiment results. We maximize following a SAA approach, sampling over realizations of $g(f(x))$. This provides the design $\hat{x}$ with the maximum posterior mean. We plot $g_{\text{true}}(f_{\text{true}}(\hat{x}))$ as the utility earned by a given PE method after a given number of pairwise comparisons.

Figure 2 shows results. EUBO-ζ and EUBO-$\tilde{f}$ perform at least as well as baseline strategies across all
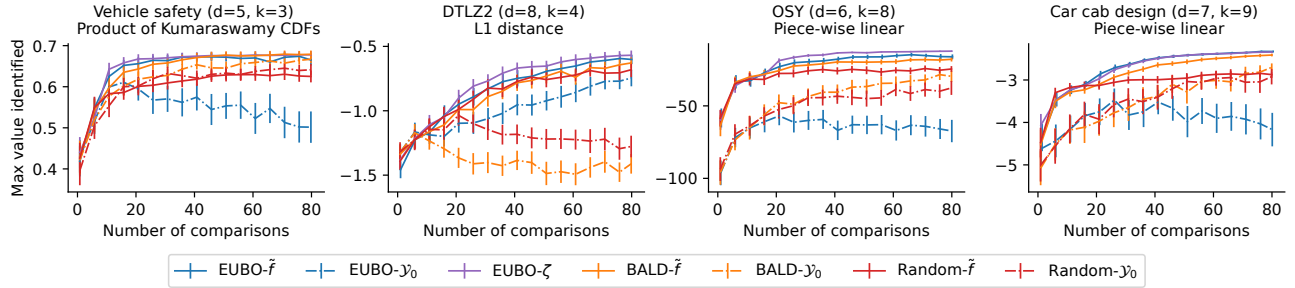
Figure 2: Mean utility of designs chosen according to maximum posterior predictive mean after a given number of pairwise comparisons during the first stage of preference exploration. CIs are ± 2 standard errors of the mean across 100 simulation replications.
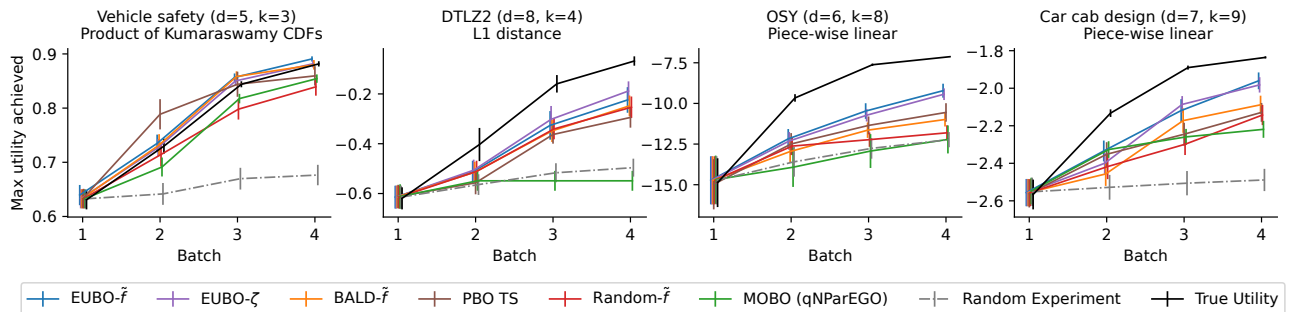

Figure 3: Max utility achieved by interleaving batches of preference exploration and experimentation. "True Utility" shown in black represents an approximate upper bound on the performance achievable obtained via Bayesian optimization with known utility. CIs are ± 2 standard errors of the mean across 30 simulation replications.

test problems on anytime performance, while BALD-$\tilde{f}$ achieves competitive but slightly inferior performance. EUBO-$\mathcal{Y}_0$ tends to over-explore outcomes that are not achievable under $f_{\text{true}}(\mathcal{X})$, leading to models of g that are progressively less accurate estimates of the posterior maximizer of $g_{\text{true}}(f_{\text{true}}(\cdot))$.

5.2 BOPE with Multiple PE Stages

We evaluate our proposed methods in a full BOPE loop with alternating rounds of experimentation and PE. After each experimentation stage, a PE stage is performed using a PE strategy. The learned preference model is then used to select designs for subsequent experiments. Each PE stage elicits 25 pairwise comparisons from the DM. This occurs over 3 rounds of experimentation, leading to a total of 75 comparisons.

Following the previous subsection, the first experimentation stage uses 32 points generated with a Sobol sequence for higher-dimensional outcome functions ($d > 5$), and 16 otherwise. For each subsequent batch, PE is performed, a batch of 16 (or 8 for the vehicle safety problem) design points are generated for each subsequent experimentation stage (i.e., batches 2 - 4). PE strategies are used in combination with qNEIUU to select designs in batches 2-4. In addition to high-

performing baselines examined in §5.1, we include a few additional baselines.

We adapt PBO to the BOPE setting as follows: For PE, we repeatedly apply PBO AFs to the results of previous experiments to elicit DM's responses over previously observed outcomes: (y_i, y_j) with $y_i, y_j \in \{y : (x, y) \in \mathcal{D}_n\}$. A standard GP model with a Laplace approximation is used to directly model the latent objective value; i.e., the mapping $x \mapsto g_{\text{true}}(f_{\text{true}}(x))$. We consider an adaptation of Thompson sampling, a popular AF used in the PBO literature (Siivola et al., 2020). This strategy is performed by selecting $x \in \mathcal{D}_n$ with the value based on independent samples from the posterior distribution of the latent objective. We refer to this strategy as PBO TS. Experiments are selected via a Monte Carlo implementation of the noisy expected improvement AF (Letham et al., 2019), qNEI. We also considered a PBO AF based on EUBO, but the results were similar to those of PBO TS.

We additionally consider the multi-objective optimization algorithm qNParEGO (Daulton et al., 2020), single-objective BO with the ground truth utility function (True Utility), and a strategy where we experiment with random Sobol design points, which are approximate upper and lower bounds on a PE strategy's
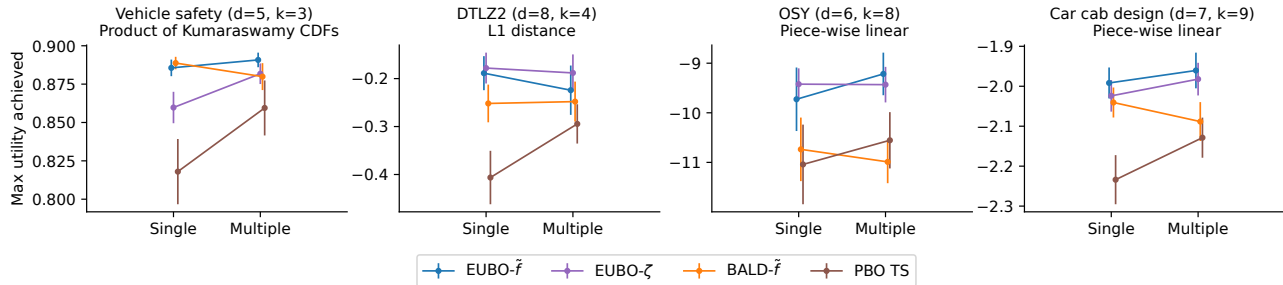
Figure 4: Max utility achieved a single round or multiple rounds of PE after the last experimentation batch. Outcomes are only marginally improved through the use of multiple PE stages for EUBO-based acquisition functions, but PBO-based strategies benefit greatly from the ability to elicit preferences across multiple rounds of experimentation. The opposite effect is seen for active learning strategies, which benefit most from conducting all learning upfront. Plots show mean and ± 2 standard errors of the mean for 30 simulation replicates.

performance. The True Utility is optimized with qNEI using a compositional objective (Astudillo and Frazier, 2019; Balandat et al., 2020).

Figure 3 shows that in all test functions presented here, BO using EUBO-ζ and EUBO-$\tilde{f}$ consistently achieves the highest utility, only second to the ground truth utility. In our experiments, we find that PBO-TS does not achieve the same level of performance as PE AFs. We also consider PE AFs that perform search directly in the entire $\mathcal{Y}_0$ domain in the SM. Similar to the previous set of experiments, search directly in $\mathcal{Y}_0$ tends to perform worse than search in $\tilde{f}(\mathcal{X})$.

5.3 BOPE with a Single PE Stage

DM interruptions are costly in practice. Therefore, we consider the case in which an initial experimentation stage takes place, followed by a single 75-comparison PE stage, after which experimentation proceeds in 3 batches of the same size as §5.2.[4]

While performing PE only once minimizes DM interruptions, it may adversely impact the optimization since all learning occurs with a surrogate of a small amount of initial random design points. Figure 4 compares the maximum utility achieved after performing four batches of experimentation, when performing all PE after the first batch, vs performing PE between rounds of experimentation. For all PE-based methods, the maximum utilities achieved are not statistically different in both settings, suggesting they are robust to different levels of PE interactivity under our experimental setting. On the other hand, we observe rather significant improvements in maximum utility achieved with multiple PE stages for PBO-TS. In this case, we see that the best value achieved quickly plateaus after the second batch of experimentation (Figure 8 in SM).

---

[4]The single PE stage approach makes it feasible to use smaller batch sizes or fully sequential optimization, but we use identical batch sizes to allow for more direct comparison.

6 CONCLUSION

BO is a prominent method for sequential experimental design, often promising to "take the human out of the loop" (Shahriari et al., 2015). However, in practice, human DMs often struggle to describe the objective they wish to optimize. This work proposes a novel human-in-the-loop BO framework with interleaved preference exploration stages called *Bayesian optimization with preference exploration*, where humans and algorithms collaborate to learn DMs' preferences over plausible outcomes for a particular black-box optimization task.

We propose EUBO, a simple and computationally efficient algorithm for exploring and learning the DM's preferences. These learned preferences in turn enable efficient search for designs whose outputs have high utilities via BO. We show that EUBO-ζ is an approximate one-step optimal policy for learning a DM's preferences with respect to our current knowledge of both the DM's utility function and the outcome function, which allows us to focus preference exploration on the most relevant parts of the outcome space with respect to the current data. EUBO provides query efficiency improvements relative to benchmark methods, finding higher utility designs while consuming less DM time and requiring fewer experiments.

This work suggests areas for future research. Real-time performance was enabled by approximate versions of the preference model and the optimal one-step optimal strategy W. State-of-the-art models such as SkewGPs (Benavoli et al., 2021), and more accurate approximations of W could potentially improve PE sample complexity while still supporting real-time interaction. BALD-$\tilde{f}$ is a strong and fast baseline, suggesting that new information-theoretic PE strategies could perform well in the BOPE framework.

Replication Material Code for replicating experiments in this paper is available at `https://github.com/facebookresearch/preference-exploration`

Acknowledgements

References

Majid Abdolshah, Alistair Shilton, Santu Rana, Sunil Gupta, and Svetha Venkatesh. Multi-objective bayesian optimisation with preferences over objectives. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 12235–12245, 2019.

Raul Astudillo and Peter Frazier. Multi-attribute bayesian optimization with interactive preference learning. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4496–4507. PMLR, 2020.

Raul Astudillo and Peter I. Frazier. Bayesian optimization of composite functions. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 354–363. PMLR, 09–15 Jun 2019.

Raul Astudillo and Peter I. Frazier. Bayesian optimization of function networks. In *Advances in Neural Information Processing Systems 34*, 2021a.

Raul Astudillo and Peter I. Frazier. Thinking inside the box: A tutorial on grey-box Bayesian optimization. In *Proceedings of the 2021 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Inc., 2021b.

Eytan Bakshy, Lili Dworkin, Brian Karrer, Konstantin Kashin, Benjamin Letham, Ashwin Murthy, and Shaun Singh. AE: A domain-agnostic platform for adaptive experimentation. *NeurIPS 2018 Systems for Machine Learning Workshop*, 2018.

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020.

Alessio Benavoli, Dario Azzimonti, and Dario Piga. Preferential bayesian optimisation with skew gaussian processes. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1842–1850, 2021.

Craig Boutilier. A POMDP formulation of preference elicitation problems. In *Proceedings of the National Conference on Artificial Intelligence*, pages 239–246, 2002.

Eric Brochu. *Interactive Bayesian optimization: learning user preferences for graphics and animation*. PhD thesis, University of British Columbia, 2010.

Eric Brochu, Nando D Freitas, and Abhijeet Ghosh. Active preference learning with discrete choice data. In J C Platt, D Koller, Y Singer, and S T Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 409–416. Curran Associates, Inc., 2008.

Roberto Calandra, André Seyfarth, Jan Peters, and Marc Peter Deisenroth. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 76(1):5–23, 2016.

Urszula Chajewska, Daphne Koller, and Ronald Parr. Making Rational Decisions using Adaptive Utility Elicitation. In *Aaai/Iaai*, pages 363—-369, 2000. ISBN 0-262-51112-6. doi: 10.1.1.34.6312.

Seok-Ho Chang, Pamela C Cosman, and Laurence B Milstein. Chernoff-type bounds for the gaussian error function. *IEEE Transactions on Communications*, 59(11):2939–2944, 2011.

Wei Chu and Zoubin Ghahramani. Preference learning with gaussian processes. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 137–144, New York, NY, USA, 2005. ACM.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. *Advances in Neural Information Processing Systems, 33*, 2020.

Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE transactions on evolutionary computation*, 18(4):577–601, 2013.

Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization*, pages 105–145. Springer, 2005.

Ian Dewancker, Michael McCourt, and Samuel Ainsworth. Interactive preference learning of utility functions for Multi-Objective optimization. December 2016.

Paul Feliot, Julien Bect, and Emmanuel Vazquez. User preferences in bayesian multi-objective optimization: the expected weighted hypervolume improvement criterion. In *International Conference on Machine Learning, Optimization, and Data Science*, pages 533–544. Springer, 2018.

Peter I Frazier. A tutorial on bayesian optimization. July 2018.

Peter I Frazier and Jialei Wang. Bayesian optimization for materials design. In *Information Science for Materials Discovery and Design*, pages 45–75. Springer, 2016.

Johannes Fürnkranz and Eyke Hüllermeier. Pairwise preference learning and ranking. In *Machine Learning: ECML 2003*, pages 145–156. Springer Berlin Heidelberg, 2003.

Javier González, Zhenwen Dai, Andreas Damianou, and Neil D Lawrence. Preferential Bayesian optimization. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1282–1291, International Convention Centre, Sydney, Australia, 2017. PMLR.

Jussi Hakanen and Joshua D Knowles. On using decision maker preferences with ParEGO. In *Evolutionary Multi-Criterion Optimization*, pages 282–297. Springer International Publishing, 2017.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. December 2011.

Shali Jiang, Daniel Jiang, Maximilian Balandat, Brian Karrer, Jacob Gardner, and Roman Garnett. Efficient nonmyopic Bayesian optimization via one-shot multi-step trees. In H. Larochelle, M. Ranzato, R. Hadsell, and M. F. Balcan, editors, *Advances in Neural Information Processing Systems 33*, pages 18039–18049, 2020.

MC Jones. Kumaraswamy's distribution: A beta-type distribution with some tractability advantages. *Statistical methodology*, 6(1):70–81, 2009.

Ponnambalam Kumaraswamy. A generalized probability density function for double-bounded random processes. *Journal of hydrology*, 46(1-2):79–88, 1980.

John R. Lepird, Michael P. Owen, and Mykel J. Kochenderfer. Bayesian preference elicitation for multiobjective engineering design optimization. *Journal of Aerospace Information Systems*, 12(10):634–645, 2015. ISSN 23273097. doi: 10.2514/1.I010363.

Benjamin Letham, Brian Karrer, Guilherme Ottoni, Eytan Bakshy, et al. Constrained bayesian optimization with noisy experiments. *Bayesian Analysis*, 14 (2):495–519, 2019.

Xingtao Liao, Qing Li, Xujing Yang, Weigang Zhang, and Wei Li. Multiobjective optimization for crash safety design of vehicles using stepwise regression model. *Structural and multidisciplinary optimization*, 35(6):561–569, 2008.

Zhiyuan Jerry Lin, Adam Obeng, and Eytan Bakshy. Preference learning for real-world multi-objective decision making. In *Workshop on Real World Experiment Design and Active Learning at ICML 2020*, 2020.

Wesley Maddox, Qing Feng, and Max Balandat. Optimizing high-dimensional physics simulations via composite bayesian optimization. *NeurIPS Workshop on Machine Learning and the Physical Sciences*, 2021.

Hongzi Mao, Shannon Chen, Drew Dimmery, Shaun Singh, Drew Blaisdell, Yuandong Tian, Mohammad Alizadeh, and Eytan Bakshy. Real-world video adaptation with reinforcement learning. *ICML 2019 Workshop on Reinforcement Learning for Real Life*, 2019.

Marcin Molga and Czesław Smutnicki. Test functions for optimization needs. *Test functions for optimization needs*, 101:48, 2005.

Andrzej Osyczka and Sourav Kundu. A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. *Structural optimization*, 10(2):94–99, 1995.

Art B. Owen. Scrambling Sobol' and Niederreiter-Xing points. *Journal of Complexity*, 14:466–489, 1998.

Daniel Packwood. *Bayesian Optimization for Materials Science*. Springer, 2017.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

Eero Siivola, Akash Kumar Dhaka, Michael Riis Andersen, Javier Gonzalez, Pablo Garcia Moreno, and Aki Vehtari. Preferential batch bayesian optimization. *arXiv preprint arXiv:2003.11435*, 2020.

Ryoji Tanabe and Hisao Ishibuchi. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89:106078, 2020.

Paolo Viappiani and Craig Boutilier. Optimal bayesian recommendation sets and myopically optimal choice query sets. In *NIPS*, pages 2352–2360, 2010.

Handing Wang, Markus Olhofer, and Yaochu Jin. A mini-review on preference modeling and articulation

in multi-objective optimization: current status and challenges. *Complex & Intelligent Systems*, 3(4): 233–245, 2017.

James Wilson, Frank Hutter, and Marc Deisenroth. Maximizing acquisition functions for bayesian optimization. In *Advances in Neural Information Processing Systems 31*, pages 9905–9916. 2018.

Yichi Zhang, Daniel W Apley, and Wei Chen. Bayesian optimization for materials design with mixed quantitative and qualitative variables. *Scientific reports*, 10(1):1–13, 2020.

Supplementary Material: Preference Exploration for Efficient Bayesian Optimization with Multiple Outcomes

A DETAILS ON ACQUISITION FUNCTIONS

A.1 Implementation of qNEIUU

Here we describe our approach to compute and optimize the qNEIUU acquisition function. Succinctly, we follow the approach of Balandat et al. (2020), which replaces the original acquisition function optimization problem with a sample average approximation (SAA). The samples used within this SAA are obtained by applying the reparameterization trick (Wilson et al., 2018) to the posterior distributions on f and g. The approximate computation of qNEIUU used within SAA is summarized in Algorithm 1. This is implemented in BoTorch (Balandat et al., 2020), and uses $N_g = 8$ samples from g samples and $N_f = 32$ samples from f. As is standard for BoTorch AFs, we use quasi-Monte Carlo samples obtained via scrambled Sobol' sequences (Owen, 1998). Optimization is performed via L-BFGS-B.

Algorithm 1 Computation of qNEIUU

Require: X, a batch of q design points to evaluate (a $q \times d$ matrix);
        X_{obs}, the set of n previously observed, potentially noisily observed points (a $n \times d$ matrix);
        f, a probabilistic surrogate model fitted on the experimental data $\mathcal{D}$ with k outcomes;
        g, a probabilistic preference model fitted on preference feedback dataset $\mathcal{P}$;
        N_f, N_g, the number of MC samples from f and g;

1: $[\tilde{Y}, \tilde{Y}_{\mathrm{obs}}] \leftarrow$ Draw N_f samples from $f([x, x_{\mathrm{obs}}])$
   # $\tilde{Y}$ is a tensor of size $N_f \times q \times k$
   # $\tilde{Y}_{\mathrm{obs}}$ is a tensor of size $N_f \times p \times k$

2: $[\tilde{U}, \tilde{U}_{\mathrm{obs}}] \leftarrow$ Draw N_g samples from $g([\tilde{Y}, \tilde{Y}_{\mathrm{obs}}])$
   # $\tilde{U}$ is a tensor of size $N_g \times N_f \times q$
   # $\tilde{U}_{\mathrm{obs}}$ is a tensor of size $N_g \times N_f \times p$

3: $U^*_{i,j} \leftarrow \max_{\ell=1,\ldots,q} \tilde{U}_{i,j,\ell}$
4: $U^*_{\mathrm{obs}\ i,j} \leftarrow \max_{\ell=1,\ldots,n} \tilde{U}_{\mathrm{obs}\ i,j,\ell}$
5: $\Delta_{i,j} := \{U^*_{i,j} - U^*_{\mathrm{obs}\ i,j}\}^+$
6: qNEIUU $\leftarrow \frac{1}{N_g N_f} \sum_{i=1:N_g} \sum_{j=1:N_f} \Delta_{i,j}$
7: **return** qNEIUU

A.2 Monte Carlo BALD

We leverage a novel MC implementation of BALD (Houlsby et al., 2011) that allows us to reparameterize the optimization of BALD in the design, rather than outcome space by propagating samples from $f(x)$ through g into in our acquisition function. The computation of this acquisition function is summarized in Algorithm 2. Here, Φ is the standard normal CDF and H_b is the binary entropy function. This acquisition function is optimized via the standard BoTorch approach described in the previous subsection.

A.3 Uniform Sampling Over $\mathcal{Y}_0$

For the Uniform Random baseline, we need to empirically determine the bounds of each outcome. To do so, we first sample a large number of random points (10^8 in this case) in $X \in \mathcal{X}$, and obtain an empirical sample of

---

Algorithm 2 MC-BALD with $\mathcal{Y}$ search space

Require: y_1, y_2, a pair of comparison design points to evaluate;
        g, a probabilistic preference model
        N_{MC}, the number of MC samples;

1: $\mu, \sigma^2 \leftarrow$ mean and var of $g(y_1) - g(y_2)$
2: $z_{\text{posterior}} \leftarrow \Phi(\frac{\mu}{\sqrt{\sigma^2+1}})$
3: $h_{\text{posterior}} \leftarrow H_b(z_{\text{posterior}})$
4: $s \leftarrow$ Draw N_{MC} samples from $N(\mu, \sigma^2)$
5: $z_{\text{samples}} \leftarrow \Phi(s)$
6: $h_{\text{conditional}} \leftarrow H_b(z_{\text{samples}}).\text{mean}()$
7: **return** $h_{\text{posterior}} - h_{\text{conditional}}$

$Y = f_{\text{true}}(X)$ by evaluating the test response function. Then we are able to identify the empirical lower and upper of f_{true} in $\mathbb{Y}$, denoted as $Y_{\min}$ and $Y_{\max}$ respectively. Finally, we scale (up or down depending on its sign and whether it's lower or upper bound) $Y_{\min}$ and $Y_{\max}$ by 10% (or 20% if one side of the bound is 0) to include additional potential boundary values that are not captured by our sampling scheme. This provides a generous uniform prior over the achievable set when used as a baselines in our experiments.

B THEORETICAL RESULTS

All probabilities and expectations in this section are with respect to the posterior distribution on g given m DM queries. Previously we used the notation $\mathbb{E}_m$, but here we drop the subindex m for brevity.

B.1 Proof of Theorem 1

Theorem 1. *Suppose that $\mathcal{Y}$ is compact, $\mu_m^g(y)$ and $K_m^g(y,y)$ are continuous functions of y, and $\lambda = 0$. Then,*

$$\underset{y_1,y_2\in\mathcal{Y}}{\operatorname{argmax}}\, \text{EUBO}(y_1,y_2) \subseteq \underset{y_1,y_2\in\mathbb{R}^k}{\operatorname{argmax}}\, V(y_1,y_2), \tag{2}$$

and the left-hand side is non-empty.

Proof. We first observe that the left-hand side is non-empty since $\mathcal{Y}$ is compact and EUBO is continuous. Continuity of EUBO follows directly from the continuity of μ_m^g and K_m^g along with (6) (note that $\Delta\Phi(\Delta/\sigma) + \sigma\varphi(\Delta/\sigma)$ is continuous at $\sigma = 0$).

Recall that

$$V(y_1,y_2) = \mathbb{E}\left[\max_{y\in\mathcal{Y}} \mathbb{E}\left[g(y) \mid (y_1,y_2,r(y_1,y_2))\right] - \max_{y\in\mathcal{Y}} \mathbb{E}[g(y)] \right].$$

Since $\max_{y\in\mathcal{Y}} \mathbb{E}[g(y)]$ does not depend on (y_1,y_2), we have

$$\underset{y_1,y_2\in\mathbb{R}^k}{\operatorname{argmax}}\, V(y_1,y_2) = \underset{y_1,y_2\in\mathbb{R}^k}{\operatorname{argmax}}\, W(y_1,y_2),$$

where

$$W(y_1,y_2) = \mathbb{E}\left[\max_{y\in\mathcal{Y}} \mathbb{E}\left[g(y) \mid (y_1,y_2,r(y_1,y_2))\right] \right].$$

Thus, it suffices to show that

$$\underset{y_1,y_2\in\mathcal{Y}}{\operatorname{argmax}}\, \text{EUBO}(y_1,y_2) \subseteq \underset{y_1,y_2\in\mathbb{R}^k}{\operatorname{argmax}}\, W(y_1,y_2). \tag{3}$$

To show (3), we rely on an idea from Viappiani and Boutilier (2010). For $i \in \{1,2\}$ let

$$y^*(y_1,y_2,i) \in \underset{y\in\mathcal{Y}}{\operatorname{argmax}}\, \mathbb{E}\left[g(y) \mid (y_1,y_2,i)\right].$$

Below, we show that

$$\text{EUBO}(y^*(y_1, y_2, 1), y^*(y_1, y_2, 2)) \geq W(y_1, y_2). \tag{4}$$

For generic y_1, y_2, we also have

$$\text{EUBO}(y_1, y_2) \leq W(y_1, y_2) \tag{5}$$

because the right-hand side is bounded below by replacing $\max_y \mathbb{E}[g(y) \mid (y_1, y_2, r(y_1, y_2))]$ by $\mathbb{E}[g(y_{r(y_1, y_2)}) \mid (y_1, y_2, r(y_1, y_2))]$ in the definition of W, i.e., by using the item that the DM selected in the query as our estimate of the item that the DM most prefers.

With these inequalities established, let (y_1, y_2) be in the right-hand side of (3). Suppose for contradiction that (y_1, y_2) is not in the left-hand side of (3). Then there must be some $(\tilde{y}_1, \tilde{y}_2)$ such that $W(\tilde{y}_1, \tilde{y}_2) > W(y_1, y_2)$.

Let $\tilde{y}_i^* = y^*(\tilde{y}_1, \tilde{y}_2, i)$ for $i = 1, 2$. We have

$$\begin{aligned}
\text{EUBO}(\tilde{y}_1^*, \tilde{y}_2^*) &\geq W(\tilde{y}_1, \tilde{y}_2) \\
&> W(y_1, y_2) \\
&\geq \text{EUBO}(y_1, y_2) \\
&\geq \text{EUBO}(\tilde{y}_1^*, \tilde{y}_2^*).
\end{aligned}$$

The first inequality is due to (4), which we show below. The second is due to our supposition for contradiction that $(\tilde{y}_1, \tilde{y}_2)$ had a strictly larger value of W than (y_1, y_2). The third is due to (5). The fourth is because (y_1, y_2) was chosen to maximize EUBO over $\mathcal{Y} \times \mathcal{Y}$ and because $\tilde{y}_i^* \in \mathcal{Y}$ for both $i = 1, 2$.

This is a contradiction. Thus, it must be that (y_1, y_2) is also in the left-hand side of (3).

It now remains to show that (4) holds. To this end, let $y_i^* = y^*(y_1, y_2, i)$. Observe that

$$W(y_1, y_2) = \sum_i \mathbb{P}(r(y_1, y_2) = i) \mathbb{E}[g(y_i^*) \mid (y_1, y_2, i)]$$

and $g(y_i^*) \leq \max\{g(y_1^*), g(y_2^*)\}$ for both $i = 1, 2$. Thus,

$$\begin{aligned}
W(y_1, y_2) &= \sum_i \mathbb{P}(r(y_1, y_2) = i) \mathbb{E}[g(y_i^*) \mid (y_1, y_2, i)] \\
&\leq \sum_i \mathbb{P}(r(y_1, y_2) = i) \mathbb{E}[\max\{g(y_1^*), g(y_2^*)\} \mid (y_1, y_2, i)] \\
&= \mathbb{E}[\max\{g(y_1^*), g(y_2^*)\}] \\
&= \text{EUBO}(y_1^*, y_2^*). \qquad \square
\end{aligned}$$

B.2   Proof of Theorem 2

Here we prove Theorem 2. First we introduce additional notation and prove several auxiliary lemmas.

Definition 1. We define

$$\text{EUBO}_\lambda(y_1, y_2) = \mathbb{E}[g(y_{r(y_1, y_2)})],$$

where $r(y_1, y_2)$ is a random variable whose conditional distribution given $g(y_1)$ and $g(y_2)$ is given by

$$\mathbb{P}(r(y_1, y_2) = 1 \mid g(y_1), g(y_2)) = \Phi\left(\frac{g(y_1) - g(y_2)}{\sqrt{2}\lambda}\right),$$

and $\mathbb{P}(r(y_1, y_2) = 2 \mid g(y_1), g(y_2)) = 1 - \mathbb{P}(r(y_1, y_2) = 1 \mid g(y_1), g(y_2))$.

Similarly, we also define

$$V_\lambda(y_1, y_2) = \mathbb{E}\left[\max_{y \in \mathcal{Y}} \mathbb{E}[g(y) \mid (y_1, y_2, r(y_1, y_2))] - \max_{y \in \mathcal{Y}} \mathbb{E}[g(y)]\right],$$

and

$$W_\lambda(y_1, y_2) = \mathbb{E}\left[\max_{y \in \mathcal{Y}} \mathbb{E}[g(y) \mid (y_1, y_2, r(y_1, y_2))]\right]$$

The above definitions generalize the definitions of EUBO, V, and W to the case where the DM's responses are corrupted by probit noise.

The following inequality is key in our proof of Theorem 2.

Lemma 1. *For any fixed $r_1, r_2 \in \mathbb{R}$ and $\lambda > 0$,*

$$\Phi\left(\frac{r_1 - r_2}{\sqrt{2}\lambda}\right) r_1 + \Phi\left(\frac{r_2 - r_1}{\sqrt{2}\lambda}\right) r_2 \geq \max\{r_1, r_2\} - \lambda C,$$

where $C = e^{-1/2}/\sqrt{2}$.

Proof. Without loss of generality we may assume that $r_1 \geq r_2$. Then, we want to prove that

$$\Phi\left(\frac{r_1 - r_2}{\sqrt{2}\lambda}\right) r_1 + \Phi\left(\frac{r_2 - r_1}{\sqrt{2}\lambda}\right) r_2 \geq r_1 - \lambda C,$$

By recalling that $\Phi(t) = 1 - \Phi(-t)$ for any $t \in \mathbb{R}$, the inequality above can be further rewritten as

$$\Phi\left(\frac{r_1 - r_2}{\sqrt{2}\lambda}\right) r_1 + \left[1 - \Phi\left(\frac{r_1 - r_2}{\sqrt{2}\lambda}\right)\right] r_2 \geq r_1 - \lambda C.$$

Arranging terms and letting $s = (r_1 - r_2)/\sqrt{2}\lambda$, we obtain the equivalent inequality

$$\frac{C}{\sqrt{2}} \geq s\left[1 - \Phi(s)\right].$$

Thus, it suffices to show that the above inequality holds for any $s \geq 0$. To this end, recall that

$$\Phi(s) = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{s}{2}\right)\right],$$

where erf is the Gauss error function. Thus, the above inequality is equivalent to

$$\frac{C}{\sqrt{2}} \geq \frac{s}{2}\operatorname{erfc}\left(\frac{s}{2}\right),$$

where $\operatorname{erfc} = 1 - \operatorname{erf}$ is the complementary Gaussian error function. Using the well-known inequality $e^{-t^2} \geq \operatorname{erfc}(t)$, which is valid for all $t > 0$ (see, e.g., Chang et al. (2011)), it suffices to show that

$$\frac{C}{\sqrt{2}} \geq \frac{s}{2}e^{-\frac{s^2}{2}};$$

i.e.,

$$\frac{1}{2}e^{-\frac{1}{2}} \geq \frac{s}{2}e^{-\frac{s^2}{2}}.$$

The above inequality can be easily verified by noting that the right-hand side reaches its maximum value at $s = 1$. $\square$

The following result shows that the expected utility of a DM expressing responses corrupted by probit noise with parameter λ is close to that of a DM expressing noiseless responses.

Lemma 2. *The following inequality holds for any $y_1, y_2 \in \mathcal{Y}$ and $\lambda > 0$:*

$$\text{EUBO}_\lambda(y_1, y_2) \geq \text{EUBO}_0(y_1, y_2) - \lambda C,$$

where C is defined like in Lemma 1.

Proof. Note that

$$\mathbb{E}[g(y_{r(y_1,y_2)}) \mid g(y_1), g(y_2)] = \Phi\left(\frac{g(y_1) - g(y_2)}{\sqrt{2}\lambda}\right) g(y_1) +$$
$$\Phi\left(\frac{g(y_2) - g(y_1)}{\sqrt{2}\lambda}\right) g(y_2).$$

Hence, from Lemma 1 it follows that

$$\mathbb{E}[g(y_{r(y_1,y_2)}) \mid g(y_1), g(y_2)] \geq \max\{g(y_1), g(y_2)\} - \lambda C.$$

The desired result can now be obtained by taking expectations over $g(y_1)$ and $g(y_2)$ on both sides of the above inequality. $\square$

Our last lemma simply shows that the function W_λ dominates EUBO_λ for any $\lambda > 0$.

Lemma 3. *The following inequality holds for any $y_1, y_2 \in \mathcal{Y}$ and $\lambda > 0$:*

$$W_\lambda(y_1, y_2) \geq \text{EUBO}_\lambda(y_1, y_2).$$

Proof. We have

$$W_\lambda(y_1, y_2) = \mathbb{E}\left[\max_{y \in \mathcal{Y}} \mathbb{E}[g(y) \mid r(y_1, y_2)]\right]$$
$$\geq \mathbb{E}\left[\mathbb{E}[g(y_{r(y_1,y_2)}) \mid r(y_1, y_2)]\right]$$
$$= \mathbb{E}[g(y_{r(y_1,y_2)})]$$
$$= \text{EUBO}_\lambda(y_1, y_2).$$

$\square$

We are now in position to prove Theorem 2.

Theorem 2. *Let $(y_1^*, y_2^*) \in \text{argmax}_{y \in \mathcal{Y}} \text{EUBO}(y_1, y_2)$. Then,*

$$V_\lambda(y_1^*, y_2^*) \geq \max_{y_1, y_2 \in \mathcal{Y}} V_0(y_1, y_2) - \lambda C,$$

where C is defined like in Lemma 1.

Proof. We have

$$W_\lambda(y_1^*, y_2^*) \geq \text{EUBO}_\lambda(y_1^*, y_2^*)$$
$$\geq \text{EUBO}_0(y_1^*, y_2^*) - \lambda C$$
$$= \max_{y \in \mathcal{Y}} \text{EUBO}_0(y_1, y_2) - \lambda C$$
$$= \max_{y \in \mathcal{Y}} W_0(y_1, y_2) - \lambda C,$$

where the first line follows from Lemma 3, the second line follows from Lemma 2, the third line follows from the definition of (y_1^*, y_2^*), and the fourth line follows from Theorem 1. The desired result can now be obtained by subtracting $\max_{y \in \mathcal{Y}} \mathbb{E}[g(y)]$ from both sides of the inequality. $\square$

B.2.1 Closed Form Expression of EUBO Under a Gaussian Posterior

To compute EUBO in closed form, recall that μ_m^g and K_m^g are the mean and posterior covariance of the approximate GP posterior after m queries. We rewrite EUBO as

$$\text{EUBO}(y_1, y_2) = \mathbb{E}[\{g(y_1) - g(y_2)\}^+ + g(y_2)]$$
$$= \mathbb{E}[\{g(y_1) - g(y_2)\}^+] + \mu_m^g(y_2).$$

Now let $\Delta(y_1, y_2)$ and $\sigma^2(y_1, y_2)$ be the mean and variance of $g(y_1) - g(y_2)$:

$$\Delta(y_1, y_2) = \mathbb{E}[g(y_1) - g(y_2)] = \mu_m^g(y_1) - \mu_m^g(y_2)$$

and

$$\sigma^2(y_1, y_2) = \mathrm{Var}[g(y_1) - g(y_2)]$$
$$= K_m^g(y_1, y_1) + K_m^g(y_2, y_2) - 2K_m^g(y_1, y_2).$$

Using the standard formula for the expectation of the positive part of a normal random variable with a given mean and variance, and dropping y_1, y_2 from the arguments to Δ and σ for brevity, we get

$$\mathbb{E}[\{g(y_1) - g(y_2)\}^+] = \Delta\Phi\left(\frac{\Delta}{\sigma}\right) + \sigma\varphi\left(\frac{\Delta}{\sigma}\right),$$

and thus

$$\mathrm{EUBO}(y_1, y_2) = \Delta\Phi\left(\frac{\Delta}{\sigma}\right) + \sigma\varphi\left(\frac{\Delta}{\sigma}\right) + \mu_m^g(y_2), \tag{6}$$

where φ and Φ are the standard normal PDF and CDF, respectively.

C SIMULATION DESIGN AND IMPLEMENTATION DETAILS

C.1 Choice of Batch Sizes and Number of DM Queries

Our simulation experiments are configured to mimic aspects of real-world experiments in which DMs may wish to perform Bayesian optimization using preference models. To do this, we rely on pilot studies related to early versions of the PE strategies developed in this work. Participants in the pilots were data scientists and ML engineers at Meta who routinely used Bayesian optimization to tune recommender system ranking policies. Standard A/B tests consider a large number of number of initial design points that are on the order of 3-5x the number of input dimensions, which motivates the initial batch sizes used in our simulation experiments. Subsequent experiments tend to use approximately half the number of points. To decide on what a sensible number of preference queries would be, we analyzed data from the pilot study to find that participants spent on average 7.8-12.7 seconds to perform pairwise comparisons between between problems with fairly large numbers of outputs (see Table 1). Based on this, we would estimate that it would conservatively take DMs around 15 minutes to perform 75 comparisons, or 5 minutes per PE stage if split across three rounds of experimentation. Finally, we observed that the empirical error rate was approximately 10%, and so we used this value for our DM noise model in the MT.

Pilot	d	k	Response mean (s)	Response sd (s)
1	11	9	10.7	5.7
2	8	9	7.8	3.0
3	4	4	8.6	6.0
4	8	6	12.7	8.1
5	8	6	8.8	7.2

Table 1: Summary statistics of input dimensionality (d), number of outcomes (k), and response times of data scientists in seconds to pairwise preference learning comparisons from pilot five studies.

D ADDITIONAL SIMULATION RESULTS

D.1 Additional Test Problems

We consider additional simulation environments here. There are four outcome and utility function combinations presented in the main paper and additional four in the supplementary material, totalling eight simulation environments. These include surrogates of real-world simulators (Vehicle Safety and Car Cab Design) and widely used synthetic functions (OSY and DTLZ2), alongside various utility functions. These test problems have 5-8 input dimensions (d), and 4-9 outcomes (k). Table 3 shows the full list of simulation environments we used in this paper. §E describe those outcome and utility in detail.

D.2 Identifying High-Utility Designs with Preference Exploration

Figure 5 Shows the experiment results for identifying high-utility designs with PE for all suite of test problems. While most results hold similar to what we observe in the main text, EUBO-ζ and EUBO-$\tilde{f}$ are performing dramatically better compared to other methods in the OSY with exponential function sum with sigmoid constraints problem. This is possibly because of the unique characteristic of this problem where many points in $\mathcal{Y}$ are violating the constraints and result in near-zero values. Comparing those near-zero utility values, while potentially helpful for learning the shape of utility functions within the achievable region, is not necessarily contributing much to the identification of the maximizer of $g_{\text{true}}(f_{\text{true}}(\mathcal{X}))$.
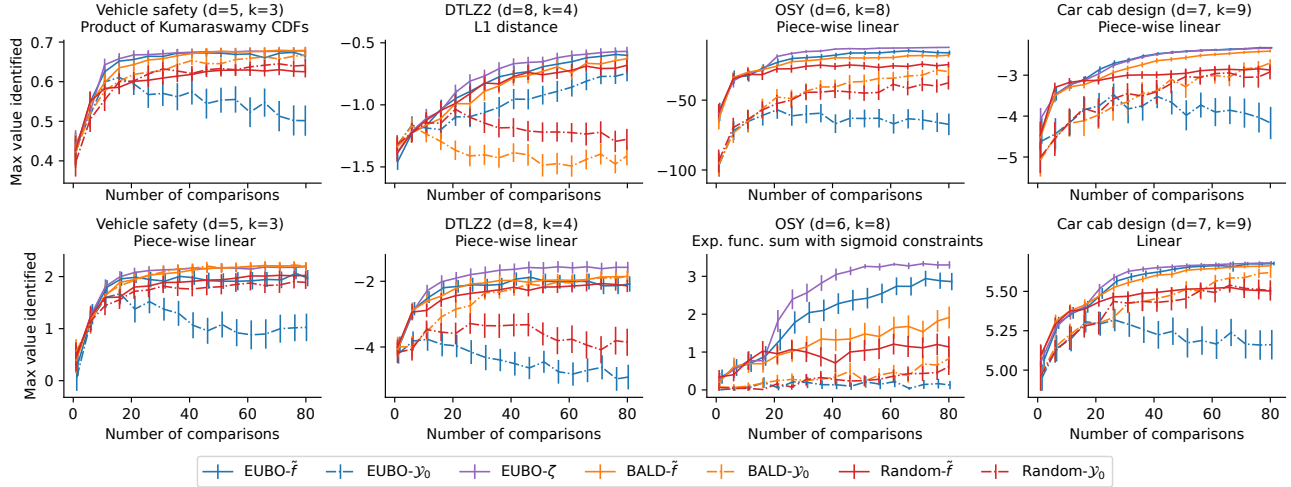

Figure 5: Simulation results for identifying high-utility designs with PE for all suite of test problems. Plotted are values of the maximum posterior predictive mean after a given number of pairwise comparisons during the first stage of preference exploration.

D.3 Preference Exploration with Multiple PE Stages

BOPE with multiple stages for additional test problems are plotted in Figure 6. Figure 6 shows the results of BOPE with multiple PE stages, and the results largely align what we observe in the main text where EUBO-ζ and EUBO-$\tilde{f}$ consistently performing well across all problems.

D.4 BOPE with One Preference Exploration Stage

Figure 7 directly shows the differences in final outcomes achieved by each method by comparing these two PE schemes. In addition to the insights from the main text, one can also see that PE AFs that perform search of $\mathcal{Y}_0$, rather than based on the posterior of f, tend to perform much better when all learning occurs upfront. This makes intuitive sense, since $\mathcal{Y}_0$ is agnostic to any additional updated surrogate models f_n collected throughout the experimentation process.

Finally, Figure 8 shows the full optimization trajectory for the single PE stage case. From these plots it is apparent that PBO-based strategies make little progress in the optimization goal after the second stage.

D.5 Probit Comparison Noise

In all simulation studies we have presented so far, we have been considering a constant 10% error rate. However, it is plausible that DMs may make errors in ways that vary with the utilities. Here, we consider the case where DMs are more likely to make mistakes when utilities have similar values, and study the behavior of the optimization strategies when such noise is present.

Concretely, we use the probit likelihood with noise level $\lambda > 0$ introduced in §3.2. Thus, when the DM is presented with a query (y_1, y_2), y_1 is chosen with probability $\Phi\left(\frac{g(y_1) - g(y_2)}{\sqrt{2}\lambda}\right)$ and y_2 is chosen with probability
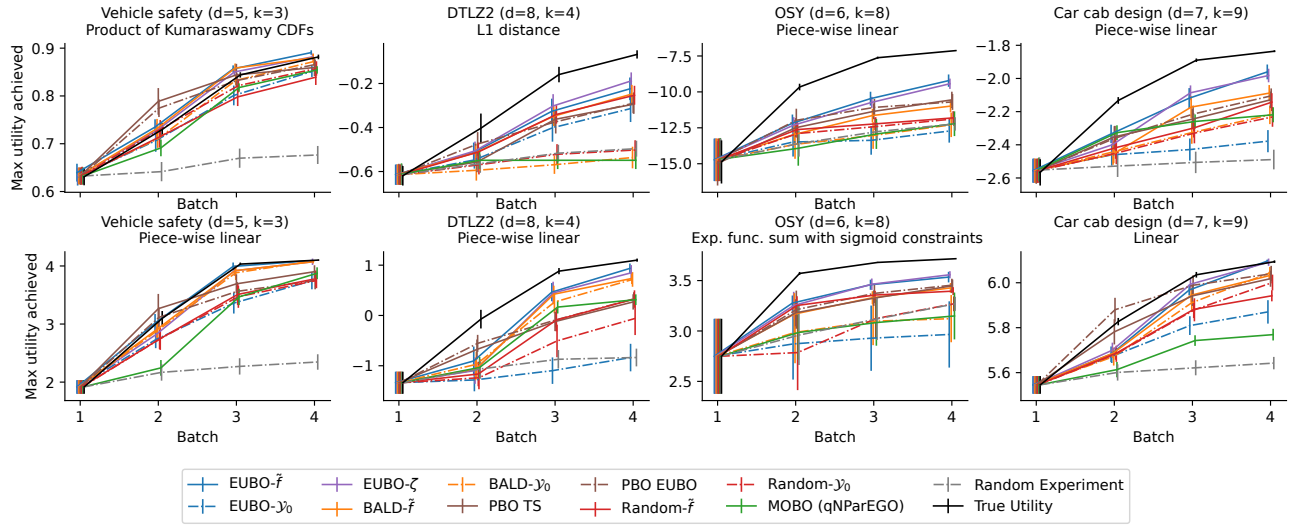
Figure 6: BOPE performance with multiple PE stages for all benchmark problems. "True Utility", the grey line on each plot, represents an approximate upper bound on the performance achievable via Bayesian optimization with a known utility function, blue and orange show our proposed methods, and other lines show competing baseline methods. The top row includes outcome and utility functions whose maximum utility achieved are plotted in the main paper in Figure 3 and the bottom row shows maximum utility achieved for the additional set of outcome and utility functions.
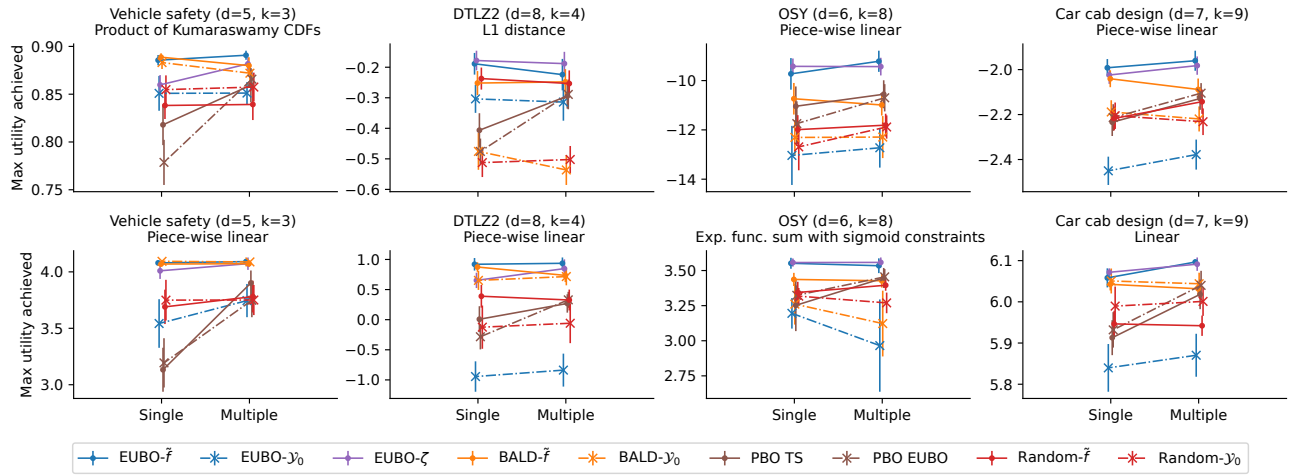

Figure 7: Maximum utility achieved after the last batch for BOPE with single PE stage and multiple PE stages for all benchmark problems.

$1 - \Phi\left(\frac{g(y_1) - g(y_2)}{\sqrt{2}\lambda}\right)$. For each experiment we set λ so that the DM makes a comparison error with probability 0.1 on average when presented with queries constituted by outcomes of designs that are in the top decile of $g_{\text{true}}(f_{\text{true}}(\mathcal{X}))$. Table 2 shows the average empirical comparison error rates observed under such probit noises.

Figure 9 shows optimization performance for BOPE with multiple PE stages for the four test problems presented in the main text using EUBO-$\tilde{f}$, EUBO-ζ, BALD-$\tilde{f}$, PBO TS, and PBO EUBO. Similar to what we observe in the main text, EUBO-ζ and EUBO-$\tilde{f}$ perform the same or better than all other PE and PBO baselines considered.
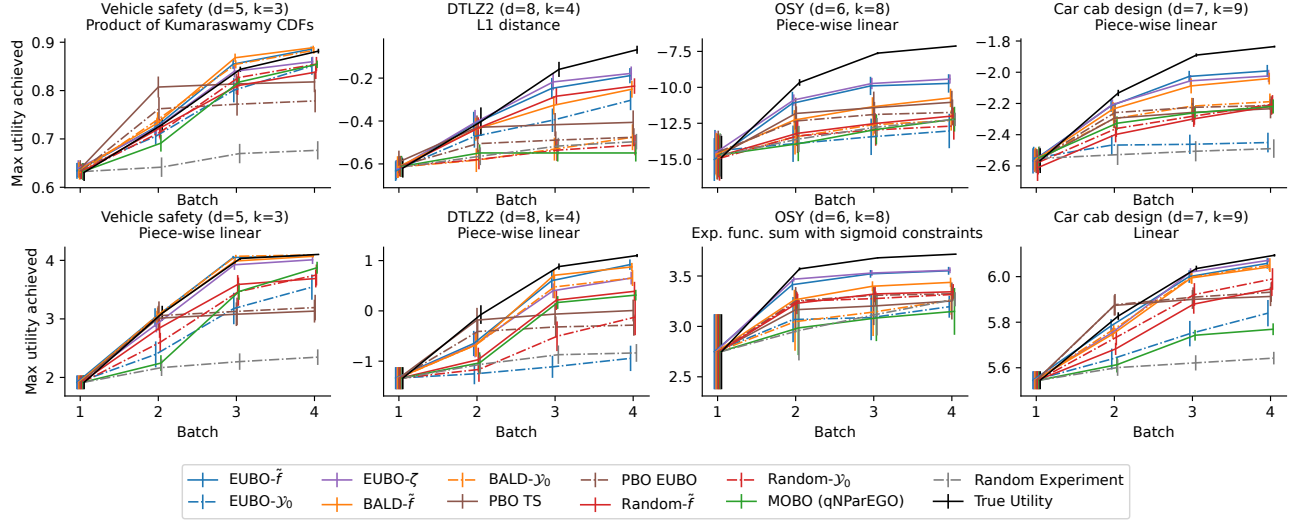

Figure 8: BOPE with a single PE stage full optimization trajectory for all benchmark problems.
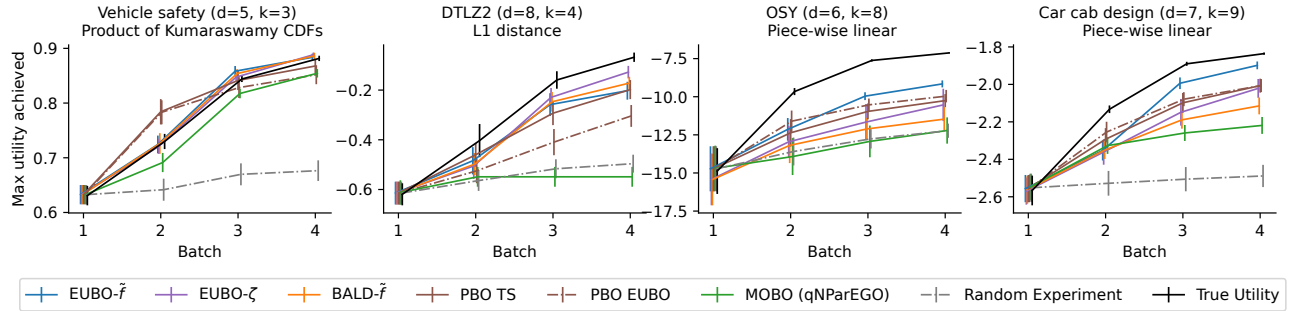

Figure 9: BOPE with multiple PE stages under probit comparison noise.

Test Problem	BALD-$\tilde{f}$	EUBO-$\tilde{f}$	EUBO-ζ	PBO EUBO	PBO TS
Vehicle safety, product of Kumaraswamy CDFs	12.8%	5.1%	10.0%	7.9%	7.8%
DTLZ2, L1 distance	3.0%	7.0%	6.9%	9.1%	4.1%
OSY, piece-wise linear	11.3%	14.0%	20.1%	15.8%	8.0%
Car cab design, piece-wise linear	12.1%	8.5%	17.9%	10.2%	7.3%

Table 2: Empirical comparison error rate under probit noise levels considered in this subsection.

E   OUTCOME AND UTILITY FUNCTIONS

In this section, we describe all outcome and utility functions used in our simulation studies.

E.1   Outcome Functions

E.1.1   DLTZ2

DTLZ2 function was first introduced by Deb et al. (2005), allowing for arbitrary input dimension d and output dimension k subject to $d > k$. $\mathbb{X} = [0,1]^d$. For a DLTZ2 function f with k-dimensional output, we have:

$$m = d - k + 1$$

$$g(x) = \sum_{i=m}^{d-1} (x_i - 0.5)^2$$

$$f_j(x) = -(1 + g(x)) \left(\prod_{i=1}^{k-j-1} \cos\left(\frac{\pi}{2} x_i\right) \right) \cdot$$

$$\mathbb{1}_{j>1} \sin\left(\frac{\pi}{2} x_{k-j-1}\right)$$

E.1.2   Vehicle Safety

This a test problem for optimizing vehicle crash-worthiness with $d = 5$ and $k = 3$. $\mathbb{X} = [1,3]^5$. We refer the readers to Tanabe and Ishibuchi (2020); Liao et al. (2008) for details on function definition. During the simulation, we normalize each component of f to lie between 0 and 1.

E.1.3   Car Cab Design

We refer the readers to Deb and Jain (2013); Tanabe and Ishibuchi (2020) for details. Note that in the original problem, there are stochastic components which we exclude in the experiments to obtain deterministic ground-truth outcome function. During the simulation, we normalize each dimension of f to between 0 and 1.

E.1.4   OSY

We adapted the constrained optimization OSY problem (Osyczka and Kundu, 1995) into a multi-objective problem by treating all constraints as objectives. We additionally flipped the signs of the two objectives of OSY such that all outputs are intended to be maximized. This adaptation makes OSY to be an outcome function with 6-dimensional inputs and 8-dimensional outputs.

E.2   Utility Functions

We consider several utility functions to capture several types of ways in which DMs may weigh the observed objective values. For all outcome functions, we consider piece-wise linear functions to represent constraint-like behavior and decreasing marginal returns. We then designed four separate utility functions by taking the characteristics of each individual test outcome function. Here we describe them in detail.

E.2.1   Piecewise Linear Function

We performed experiments on all test outcome functions using piece-wise linear functions as their shapes correspond to real-world diminishing marginal returns on outcomes and sharp declines in utility once constraints are violated. For a k-dimensional input vector $\mathbf{y}$, this utility function g is defined as

$$g(\mathbf{y}; \beta_{\mathbf{1}}, \beta_{\mathbf{2}}, \mathbf{t}) = \sum_{i=1}^{k} h_i(y_i)$$

where

$$
h_i\left(y_i\right) =
\begin{cases}
\beta_{1,i} \cdot y_i + (\beta_{2,i} - \beta_{1,i})t_i & y_i < t_i \\
\beta_{2,i} \cdot y_i & y_i \geq t_i
\end{cases}
$$

For DTLZ2 (d=8, k=4) problem, we set

$$
\begin{aligned}
\beta_1 &= [4, 3, 2, 1] \\
\beta_2 &= [0.4, 0.3, 0.2, 0.1] \\
\mathbf{t} &= [1, 1, 1, 1].
\end{aligned}
$$

For vehicle safety problem, we set

$$
\begin{aligned}
\beta_1 &= [2, 6, 8] \\
\beta_2 &= [1, 2, 2] \\
\mathbf{t} &= [0.5, 0.8, 0.8].
\end{aligned}
$$

For the car cab design problem, we set

$$
\begin{aligned}
\beta_1 &= [7.0, 6.75, 6.5, 6.25, 6.0, 5.75, 5.5, 5.25, 5.0] \\
\beta_2 &= [0.5, 0.4, 0.375, 0.35, 0.325, 0.3, 0.275, 0.25, 0.225] \\
\mathbf{t} &= [0.55, 0.54, 0.53, 0.52, 0.51, 0.5, 0.49, 0.48, 0.47]
\end{aligned}
$$

and the threshold parameter $\mathbf{t_i} = 0.75$ for all i.

For the OSY problem, we set

$$
\begin{aligned}
\beta_1 &= [0.02, 0.2, 10, 10, 10, 10, 10, 10] \\
\beta_2 &= [0.01, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1] \\
\mathbf{t} &= [1000, -100, 0, 0, 0, 0, 0, 0].
\end{aligned}
$$

E.2.2 Linear Function

For the car cab design problem, we experiment with a linear utility function. For a k-dimensional outcomes vector $\mathbf{y}$, this utility function g is defined as

$$
g(\mathbf{y}; \beta) = \beta^T \mathbf{y}
$$

Specifically, we set

$$
\beta = [2.25, 2, 1.75, 1.5, 1.25, 1, 0.75, 0.5, 0.25].
$$

E.2.3 Product of Kumaraswamy Distribution CDFs

Prior work on preference learning has utilized the Beta CDF to form utility functions (Dewancker et al., 2016). The Beta CDF provides a convenient, bounded monotonic transform that smoothly varies between increasing and decreasing marginal gains with respect to their inputs. In this work, we utilize the Kumaraswamy CDF (Kumaraswamy, 1980; Jones, 2009), which behaves much like the Beta CDF, but is differentiable. This allows us to optimize qNEIUU via gradient ascent using the true utility function in Section 5. For the vehicle safety problem, we experiment with the product of Kumaraswamy distribution CDFs as its utility function, representing that we wish to simultaneously achieve high utility values along each individual dimension without sacrificing much on others. For a k-dimensional input vector $\mathbf{y}$, this utility function g is defined as

$$
g(\mathbf{y}; \mathbf{a}, \mathbf{b}) = \prod_{i=1}^{k} F_i\left(y_i\right),
$$

where $F_i(\cdot)$ is the CDF of a Kumaraswamy distribution with shape parameters a_i and b_i.

Specifically, we set

$$\mathbf{a} = [0.5, 1, 1.5]$$
$$\mathbf{b} = [1, 2, 3].$$

E.2.4 L1 Distance

For the DTLZ2 problem, we additionally test a utility function using negative L1 distance from a Pareto-optimal point. This choice of utility function mimics the scenario where the DM wish to keep the outcomes close to a specific desirable state such as physiological measurements in medical applications.

For the DTLZ2 problem, we choose the Pareto-optimal point to be $y^{\mathrm{PO}} = \mathrm{DTLZ2}(x^{\mathrm{PO}})$ where $x_i^{\mathrm{PO}} = 0.5$ for all i.

E.2.5 Exponential Function Sum with Sigmoid Constraints

The OSY problem is originally a constrained optimization problem and we wish design an utility function that can reflect this nature of this problem.

For a k-dimensional outcomes vector $\mathbf{y}$, we recall that the first two dimensions y_1 and y_2 are objectives we wish to maximize and in the original OSY problems, the remaining six constraints $y_{3..8}$ need to be kept positive. To reflect these goals, we first normalize the outputs of y_1 and y_2 to be between 0 and 1 using min-max normalization. Given an outcome vector $\mathbf{y}$, the utility function is then given by

$$g(y) = (\exp(y_1) + \exp(y_2)) \prod_{j=3}^{8} S\left(\frac{50 y_j}{\min\{-y_j^{\min}, y_j^{\max}\}}\right)$$

where S is the sigmoid function; y_j^{min} and y_j^{max} are empirically determined lower and upper bound of y_j.

Outcome function	Utility function	d	k	Reference
Vehicle safety	Piecewise linear function	5	3	Liao et al. (2008)
	Product of Kumaraswamy distribution CDFs			Tanabe and Ishibuchi (2020)
DTLZ2	Piecewise linear function	8	4	
	L1 distance from Pareto-optimal point			Molga and Smutnicki (2005)
OSY	Piecewise linear function	6	8	
	Exponential function sum with sigmoid constraints			Osyczka and Kundu (1995)
Car cab design	Piecewise linear function	7	9	Deb and Jain (2013)
	Linear function			Tanabe and Ishibuchi (2020)

Table 3: Complete list of outcome and utility function combinations. d and k refers to input and output dimension of the test outcome function respectively.