Consecutive Preferential Bayesian Optimization

Anonymous authors
Paper under double-blind review

Abstract

Preferential Bayesian optimization allows optimization of objectives that are either expensive or difficult to measure directly, by relying on a minimal number of comparative evaluations done by a human expert. Generating candidate solutions for evaluation is also often expensive, but this cost is ignored by existing methods. We generalize preference-based optimization to explicitly account for production and evaluation costs with *Consecutive Preferential Bayesian Optimization*, reducing production cost by constraining comparisons to involve previously generated candidates. We also account for the perceptual ambiguity of the oracle providing the feedback by incorporating a *Just-Noticeable Difference* threshold into a probabilistic preference model to capture indifference to small utility differences. We adapt an information-theoretic acquisition strategy to this setting, selecting new configurations that are most informative about the unknown optimum under a preference model accounting for the perceptual ambiguity. We empirically demonstrate a notable increase in accuracy in setups with high production costs or indifference feedback, including a real-world task in food science.

1 Introduction

Preferential Bayesian optimization (PBO) tells how to efficiently find the optimum of an unknown function $f(y) \in \mathbb{R}$ when we cannot even evaluate the function itself but can only conduct *preferential queries* (Chu & Ghahramani, 2005; González et al., 2017). That is, we only have an oracle (often a human expert) for evaluating the relative quality of two candidates y_i and y_j, which must be used to learn a proxy for $f(y)$. As a concrete example, in an algorithmic search for the best possible avocado toast recipe, we want to leverage preferential comparisons since providing absolute scores is difficult (Kochanski et al., 2017), and we want to minimize the number of preferential queries that all require eating a toast. However, most real optimization tasks involve another cost that the PBO literature ignores: Making the toasts is not free either. Methods like the popular qEUBO (Astudillo et al., 2023) that query preferences over *multiple* new candidates can be especially wasteful in this common setting.

We introduce *Consecutive Preferential Bayesian Optimization* (CPBO), a new problem formulation for addressing preferential optimization setups where the cost of producing the candidates is not negligible (Fig. 1A-B). Instead of freely proposing candidates for new preferential queries, we now need to consider as reference points some candidates that have already been produced. Whenever the cost of producing a candidate is sufficiently large, an optimal strategy is to consider *only* preference queries of the type $y_t \succ y_l$, where y_l is a candidate solution produced during an earlier iteration $l < t$. We introduce the general setup, explain how the performance of current PBO methods changes when constrained to operate consecutively, even though they were not originally designed for it, and discuss how the relative costs of production and evaluation influence the optimal choices.

Within the general CPBO setup, we focus on a case with an additional constraint: Instead of all previously produced candidates, we can only compare y_t to y_{t-1}, the previous one. This is needed when the candidates deteriorate over time: Avocados need to be consumed fresh, and in our concrete empirical experiment of high-moisture extrusion processing (Section 4.3) the candidates need to be evaluated while they are still warm after production (Lawless & Heymann, 2010). Conditions may also need to be experienced consecutively to provide a reliable assessment, as in thermal comfort optimization (Xu et al., 2024). We propose a concrete

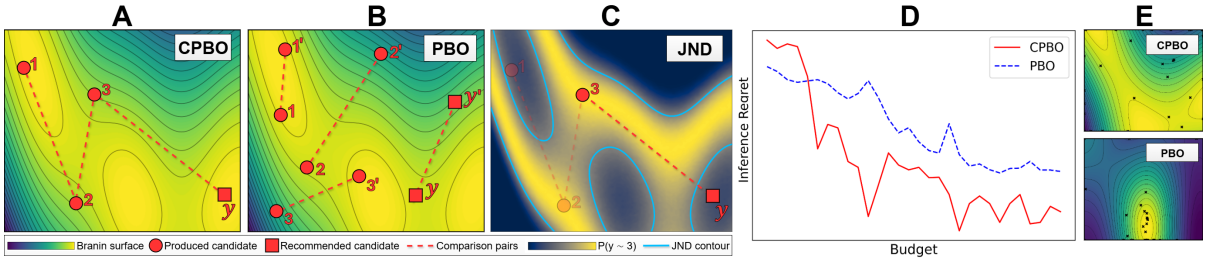

Figure 1: Consecutive Preferential BO (CPBO) operates with consecutive comparisons (A), in contrast to standard PBO where two (or more) new candidates are proposed at each iteration (B). For each comparison we model how likely the expert cannot tell the candidates apart; (C) shows this probability as a heatmap when candidate is compared to candidate "3", with the contour marking the Just-Noticeable Difference (JND) threshold. Under non-zero production costs and JND, CPBO improves both optimization speed (D) and the quality of the learned utility proxy (E) over standard PBO, here EUBO by Lin et al. (2022).

algorithm that extends Max-value Entropy Search (MES) (Wang & Jegelka, 2017) to select the optimal candidate to be produced and compared against the previous candidate, representing the first practical solution for automating optimization for this type of problem. Even though Xu et al. (2024) mentioned the setup anecdotally, they did so only to simplify theoretical analysis and did not consider any real consecutive optimization problems.

Setups integrally relying on human evaluations warrant more careful consideration of the evaluation aspect. The PBO literature is primarily motivated by tasks where the oracle is a human expert – for example, in animation design (Brochu et al., 2010) or in molecular configuration (Mikkola et al., 2020) – but most works model humans as perfect oracles, assuming they have definite answers to all queries: either $y_i \succ y_j$ or $y_i \prec y_j$. That is, the evaluator is forced to prefer one toast over the other, even when they regard them as equally good or when there are conflicting signals (*"I prefer the spices in this one, but the toast is less crusty."*). This oversimplification has been questioned in other learning setups, with demonstrated improvement from accounting for the indifference in human preferences (Liu et al., 2024).

We address this shortcoming using the concept of *Just-Noticeable Difference* (JND) (Weber, 1978). In addition to $y_i \succ y_j$ or $y_i \prec y_j$, we also consider $y_i \sim y_j$, where $\sim$ is the commonly used symbol for *indifference* to the preference query. Indifference-aware preference modeling has been considered previously in the context of standard preference-based BO. Nguyen et al. (2021) introduce a learnable indifference threshold within the Plackett–Luce model (Plackett, 1975; Luce, 1959) and propose an entropy-based acquisition strategy for top-k ranking problems, which reduces to the Bradley–Terry model (Bradley & Terry, 1952) in the case of pairwise comparisons. Their setting assumes that arbitrary candidate pairs or subsets can be freely selected at each iteration. In contrast, we focus on a cost-constrained consecutive setting, where only one new candidate is generated per iteration and comparisons must be made against previously produced references, fundamentally changing the acquisition problem and precluding a direct application of their method. We model such ambiguity by extending the Thurstone–Mosteller model (Thurstone, 1927; Mosteller, 1951) with a learnable indifference band in a latent space underlying the assumed utility. We empirically compare the proposed CPBO method against concurrent PBO methods (Lin et al., 2022; Xu et al., 2024) in scenarios of varying degrees of indifference, showing that already for cases where $10 - 20\%$ of comparisons are indicated as indifferent, our method clearly outperforms them (Fig. 1C-E). Furthermore, having some indifferences actually helps, compared to assuming perfect answers for all queries; Fig. 1C shows how the JND helps avoid querying candidates with similar expected utility. In this sense, the consecutive setting and JND are not independent ingredients: the consecutive constraint fixes one side of the comparison, which makes the JND threshold directly actionable when selecting the next candidate. In particular, the acquisition rule can avoid comparisons whose expected utility difference is likely below the JND, while also avoiding overly obvious comparisons that provide little information. Thus, CPBO provides a natural setting in which perceptual ambiguity can influence the optimization loop itself, rather than only being handled retrospectively in the preference model.

We make the following concrete contributions to the PBO literature. **Extended cost-awareness:** We explain how PBO needs to account also for the cost of producing the candidates, not just the evaluation, and illustrate the production-evaluation cost balance. **Consecutive PBO:** We characterize a new learning setup, where comparisons are constrained to previously produced candidates to minimize the production cost. **Practical Method:** We extend the Thurstone model to account for JND and adapt MES for conditional comparisons, arriving at a concrete CPBO method for real optimization tasks. **Experimental Validation:** Quantification of the importance of accounting for indifference evaluations, and a real-world laboratory demonstration in food extrusion optimization using preference data physically collected under real operational constraints.

2 Problem Setup

Cost-Aware Preferential BO. Analogously to the setup presented by Lin et al. (2022), proper treatment of any PBO setup requires three quantities:

x: A configuration to be evaluated (recipe for a toast; settings of an extruder)

$y(x)$: The candidate itself, produced with configuration x (an actual toast; extrudate)

$f(y)$: The utility of the user preference, so that $f(y) > f(y')$, denoted by $y \succ y'$, implies the user prefers candidate y over y' (taste; subjective evaluation of overall quality)

Our goal is to find the configuration x that maximizes the utility for the corresponding candidate y as $\arg\max_x f(y(x))$, by selecting a sequence of configurations $x_1, .., x_T$ while respecting a budget $B > 0$ such that $\sum c_p + \sum c_e \leq B$. Here, each configuration incurs the *production cost* $c_p \geq 0$ to produce $y(x)$, and each pairwise comparison (x, x') incurs the *evaluation cost* $c_e \geq 0$ to evaluate $\mathbb{I}(y(x) \succ y(x'))$. The sums are over *unique* productions and judgments: the production cost incurs only once per unique candidate, not when using the candidate again in a later comparison.

The existing PBO literature does not explicitly distinguish between the production cost c_p and the evaluation cost c_e, lacking tools to account for them. Our formulation makes three separate scenarios apparent:

1. For $c_p \ll c_e$, we can ignore the production cost, as implicitly done by previous PBO methods tracking progress in terms of preference evaluations (González et al., 2017). qEUBO (Astudillo et al., 2023) takes this to the extreme, proposing $q \geq 2$ new candidates for a single evaluation.

2. For $c_p \approx c_e$, we should typically produce only one new candidate for each comparison, selecting the other candidate from those already produced to avoid a new production cost.

3. For $c_p \gg c_e$, each newly produced candidate should be compared against *all* previous ones. For instance, in the duel bandit context, Agarwal et al. (2022) adds every new candidate to a candidate set and uses all pairwise comparisons.

Consecutive Preferential BO. The first case is well covered by current methods and addressing the third one requires only a minor extension using multiple comparisons. Our main focus is in the unexplored middle case. The costs, in general, can be addressed by the standard formulation of cost-aware BO (Swersky et al., 2013; Lee et al., 2020; Luong et al., 2021), but we need one new component: How to select an optimal x_j conditional on already fixed x_i.

To focus on the novel aspect, we present all details for the special case where we *only* consider such conditional acquisitions. Furthermore, for notational simplicity, we limit the reference point to be the previous candidate y_{t-1}. In scenarios where the candidates deteriorate over time, it is the only candidate that can be assessed without incurring yet a third possible cost, the *retrieval cost* c_r for storing (e.g., freezing) and retrieving (de-frosting) historical candidates. This special case is a direct answer to a concrete need, as demonstrated in Section 4.3.

The extensions for the general case are straightforward. For instance, if L previous candidates can be assessed without the retrieval cost, we can compute the information gains for all L choices in parallel. Similarly, if $c_p \approx c_e$, we can compute information gain for both a standard proposal of two new candidates and for a conditional proposal, selecting the one with the higher gain per cost unit.

3 Method

To solve the CPBO problem, we propose a preferential BO algorithm with three key components. First, it uses consecutive comparisons as defined above. Second, it models preference feedback using a Random Utility Model (RUM) (McFadden, 1974; Manski, 1977) with a JND threshold to capture indifference. Third, it selects the next candidate y by maximizing the mutual information between the preference response and the unknown maximum of the latent utility proxy $f(x)$, using an adapted variant of MES. As a proxy, we make the common choice of a Gaussian Process (GP) (Williams & Rasmussen, 2006), with analytic expressions for many required quantities.

3.1 User Model

We model user feedback using a RUM, where preferences arise from latent utility values corrupted by perceptual noise. At each iteration, a new candidate $y_t = y(x_t)$ is produced with a configuration $x_t \in \mathcal{X}$. The expert compares this to the previous candidate y_{t-1} and provides a discrete preference label $R_t \in \mathcal{R} = \{-1, 0, +1\}$, indicating the preferred candidate or indifference. We denote its observed realization by r_t.

After t iterations, the preference data is $\mathcal{D}_t = \{([x_{i-1}, x_i], r_i)\}_{i=2}^{t}$ with $y_i = y(x_i)$. Although preferences are given for the physical outputs $y(x)$, we can place our latent utility directly on configurations by defining $f(x)$ to be the utility of $y(x)$, treating the production as deterministic. This lets us learn $f(x)$ over $\mathcal{X}$ without an explicit model of y, but an extension for stochastic $y(x)$ would be straightforward.

To model uncertainty over user preferences, we place a GP prior on the latent utility: $f \sim \mathcal{GP}(\mu, k)$, where the prior mean is constant, $\mu(x) = m$, and the covariance is a radial basis function (RBF) kernel with automatic relevance determination (ARD), where each ℓ_d is a separate lengthscale for input dimension d:

$$k(x, x') = \sigma_f^2 \exp\Big(-\tfrac{1}{2} \sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{\ell_d^2}\Big). \tag{1}$$

The GP output scale σ_f^2 is fixed to remove the scale ambiguity discussed below. We follow the standard preferential GP inference framework (Chu & Ghahramani, 2005), but replace Laplace approximation with variational inference. However, our likelihood extends standard preferential GP formulations by explicitly modeling JND. Specifically, we form our preference likelihood by extending the classical Thurstone–Mosteller model (Thurstone, 1927; Mosteller, 1951) to a three-outcome setting with an explicit indifference region. Indifference-aware extensions of random utility models have been studied in the preference learning literature (Davidson, 1970; Zhou et al., 2008), and Benavoli & Azzimonti (2024) also discusses JND in GP preference learning. Our use of JND differs in that we combine the threshold with perceptual noise to obtain smooth response probabilities for the three possible outcomes, and use these probabilities inside the acquisition function for consecutive Bayesian optimization, introducing modifications required for stable inference and effective acquisition.

We assume that the expert implicitly evaluates a latent utility difference

$$\Delta f_t = f(x_t) - f(x_{t-1}), \tag{2}$$

but perceives only noisy utilities. Following the Thurstone-Mosteller model, we assume additive Gaussian noise on each candidate, $\varepsilon_t, \varepsilon_{t-1} \sim \mathcal{N}(0, \sigma^2)$, making the perceived utility difference

$$\Delta f_t + \delta_t, \quad \text{where} \quad \delta_t = \varepsilon_t - \varepsilon_{t-1} \sim \mathcal{N}(0, 2\sigma^2). \tag{3}$$

The user's response compares the noisy utility difference within the JND threshold $\gamma > 0$

$$R_t = \begin{cases} +1, & \text{if } \Delta f_t + \delta_t > \gamma, \\ 0, & \text{if } |\Delta f_t + \delta_t| \leq \gamma, \\ -1, & \text{if } \Delta f_t + \delta_t < -\gamma. \end{cases} \tag{4}$$

This threshold γ models perceptual ambiguity, mapping small utility differences more likely to indifference. For $\gamma = 0$, it reduces to standard binary preferences (Chu & Ghahramani, 2005).

Given the utility function estimate f, the likelihood of a single preference observation can be written as,

$$\begin{aligned} P(R_t = +1 \mid f) &= \Phi\left(\tfrac{\Delta f_t - \gamma}{\sqrt{2}\sigma}\right), \\ P(R_t = 0 \mid f) &= \Phi\left(\tfrac{\gamma - \Delta f_t}{\sqrt{2}\sigma}\right) - \Phi\left(\tfrac{-\gamma - \Delta f_t}{\sqrt{2}\sigma}\right), \\ P(R_t = -1 \mid f) &= \Phi\left(\tfrac{-\Delta f_t - \gamma}{\sqrt{2}\sigma}\right), \end{aligned} \tag{5}$$

where Φ denotes the standard normal CDF. The joint likelihood of the data is hence

$$P(\mathcal{D}_t \mid f) = \prod_{i=2}^{t} P(r_i \mid f), \tag{6}$$

with the posterior $p(f|\mathcal{D}_t) \propto P(\mathcal{D}_t \mid f)p(f)$, where $p(f)$ is the GP prior.

Details. We use variational approximation to estimate the posterior (Titsias, 2009), implemented with the `GPyTorch` library (Gardner et al., 2018). At iteration t, let $\mathbf{u} = [f(x_1), \ldots, f(x_t)]^\top$ denote the latent utility values at all configurations observed so far. We define a multivariate Gaussian variational distribution $q(\mathbf{u}) = \mathcal{N}(\mathbf{m}_u, \mathbf{L}\mathbf{L}^\top)$, where $\mathbf{m}_u \in \mathbb{R}^t$ and $\mathbf{L} \in \mathbb{R}^{t \times t}$ is a Cholesky factor. The approximate posterior over the latent utility function is then obtained by integrating the GP conditional,

$$q\big(f(\cdot)\big) = \int p\big(f(\cdot) \mid \mathbf{u}\big) q(\mathbf{u})\, d\mathbf{u}. \tag{7}$$

We jointly optimize the variational parameters, the constant prior mean, the ARD lengthscales, and the JND threshold γ by maximizing the Evidence Lower Bound (ELBO) as an approximate marginal likelihood. The expected log-likelihood term in the ELBO is estimated by Monte Carlo samples of latent utilities drawn from $q(f)$, and the resulting objective is optimized with gradient-based methods.

Due to the latent structure of the model, where feedback depends only on utility differences, the solution is not identifiable. The likelihood is invariant under joint rescaling of the latent function $f(x)$ and the perceptual noise σ; $\Delta f/\sigma$ and γ/σ are invariant to scaling. To remove this invariance, we fix σ and the GP output scale to constant reference values. Importantly, this choice does not affect optimization behavior: as shown in the supplement, performance is robust over a wide range of σ values (spanning more than an order of magnitude), yielding statistically indistinguishable results (see Supplement C.4). Further details on the model and computation, including numerical stabilization and optimization settings, are provided in Supplement B.1.

3.2 Acquisition Function

We adapt Max-value Entropy Search (MES) (Wang & Jegelka, 2017) to the CPBO setting, where each new candidate $y(x)$ is compared against the previous one y_t. The goal is to select the next configuration x that maximizes the mutual information with the unknown maximum value

$$f^* = \max_{x \in \mathcal{X}} f(x), \tag{8}$$

based on the expert's next preference feedback R_{t+1}. We define the acquisition function as

$$\begin{aligned} \alpha(x) &= I(f^*; R_{t+1} \mid \mathcal{D}_t) \\ &= H(R_{t+1} \mid \mathcal{D}_t) - \mathbb{E}_{f^*}\left[H(R_{t+1} \mid f^*, \mathcal{D}_t)\right], \end{aligned} \tag{9}$$

where the expectation is over $p(f^* \mid \mathcal{D}_t)$ and H denotes entropy. In our setting, this entropy is over the three possible responses $R_{t+1} \in \{-1, 0, +1\}$, with response probabilities computed from the JND likelihood in Eq. (5).

Approximation. Following the original MES (Wang & Jegelka, 2017), we discretize $\mathcal{X}$ into a candidate set $\hat{\mathcal{X}}$ and approximate the maximum utility $f^* = \max_{x \in \hat{\mathcal{X}}} f(x)$ by fitting a Gumbel distribution to quantiles estimated from the GP posterior. The unconditional entropy term $H(R_{t+1} \mid \mathcal{D}_t)$ is estimated using Monte Carlo samples from the GP posterior, with predictive response probabilities computed using Eq. (5). For each $f_j^* \sim$ Gumbel, we construct a truncated GP posterior $p_{\leq f_j^*}$ conditioned on $\mathcal{D}_t$, where the truncation enforces $\max\{f(x_t), f(x)\} \leq f_j^*$.

The conditional feedback distribution is then estimated as

$$P(R_{t+1} = r \mid f_j^*, \mathcal{D}_t) = \mathbb{E}_{f \sim p_{\leq f_j^*}} \left[P(r \mid \Delta f) \right], \tag{10}$$

where $\Delta f = f(x) - f(x_t)$ and $P(r \mid \Delta f)$ is given by Eq. (5). Acquisition values are computed via Monte Carlo integration over Gumbel samples. In practice, directly evaluating the expectation over f^* would require constructing a separate truncated posterior for many sampled maximum values. To reduce this cost, we sample maximum values from the fitted Gumbel distribution, group them into equally spaced bins, and use the mean value within each bin as a representative truncation level. This reduces the number of truncated posterior computations while retaining a Monte Carlo approximation of the expectation over f^*.

The acquisition function can be optimized in several ways. To solve the stochastic optimization problem caused by the Monte Carlo estimate of the conditional entropy, we use a gradient-free optimizer, specifically Bayesian optimization for this inner acquisition function optimization loop. Although one could employ a gradient-based optimizer utilizing automatic differentiation and the reparameterization trick (e.g., as in the BoTorch framework Balandat et al. (2020)), such methods frequently necessitate multi-start gradient optimization. Using Bayesian optimization as a global optimizer avoids this issue. For numerical stability, we apply zero-mean normalization before computing entropy terms: for each posterior utility sample $f^{(s)}$ and evaluation batch $\mathcal{X}_{\text{eval}} = \{x^{(1)}, \ldots, x^{(N)}\}$, we subtract the empirical mean $N^{-1} \sum_{j=1}^{N} f^{(s)}(x^{(j)})$ from every value of $f^{(s)}$ in that batch. This does not change the preference probabilities, since the likelihood in Eq. (5) depends only on utility differences.

Compared to standard MES, our acquisition function introduces the following modifications:

Consecutive comparisons: Each comparison is against the previous candidate.

Three-outcome feedback: Entropies are computed over $P(r \mid f)$ given by Eq. (5).

Local dual truncation: We condition the GP posterior jointly on x and x_t using f_j^*.

Supplement B.2 provides additional details, including the Gumbel fitting equations and the use of the posterior median for centering the fitted distribution, as well as binning parameters, zero-mean normalization of posterior samples, and numerical stabilization.

4 Experiments

We empirically verify our two main claims. First, we show how the balance between the production and evaluation costs influences the general pattern of PBO, using the EUBO method (Lin et al., 2022) in different ways. We perform this experiment with an established standard PBO method, rather than our proposal, to draw attention to the general findings. Second, we show strong overall performance of our method in the CPBO setup with significant improvements over the baselines in the presence of JND. Finally, we cast a recurring optimization task in food science as CPBO, demonstrating the method in a real case where consecutive comparisons are the only information source.

Across all synthetic benchmarks, we use a fixed algorithmic configuration without manual hyperparameter tuning. GP hyperparameters are learned during the algorithm or fixed to reference values, and the only

adjustable setting is the acquisition optimization budget. The same configuration is used in the real-world extrusion case study, with a weak prior on GP lengthscales for stabilization (see Supplement D.5).

Baselines. We compare against two recent PBO methods: EUBO by Lin et al. (2022) as implemented in the `BoTorch` library (Balandat et al., 2020), and POP-BO by Xu et al. (2024) as implemented in the authors' code release. For EUBO, we adapt the query strategy by forcing the first alternative to be the candidate proposed in the previous query, and optimizing only over the second alternative. Note that EUBO is similar but not identical to qEUBO with $q = 2$ (Astudillo et al., 2023); in Supplement C.4 we show additional results with qEUBO.

POP-BO natively supports consecutive comparisons but requires direct evaluations of $f(y)$ to determine various hyperparameters and, hence, it can not be used in pure PBO tasks. To side-step this limitation, we re-use the hyperparameter choices Xu et al. (2024) provided using such evaluations, and exclude POP-BO for functions they did not consider (NAs in Table 1). Nonetheless, we emphasize that this gives POP-BO an unfair advantage over both CPBO and EUBO.

Metrics. We use the standard BO evaluation metrics: *inference regret* $f(x^*) - f(\hat{x}_n)$, where $\hat{x}_n = \arg\max_{x \in \mathcal{X}} \mu_n(x)$ is the model's final recommendation, and *simple regret*, which measures the quality of the best solution thus far as $f(x^*) - \max_{1 \leq i \leq n} f(x_i)$. To assess the level of exploration, we measure the global quality of the utility proxy $f(x)$ after optimization using *ordinal accuracy*, which calculates binary ranking consistency over uniformly random configuration pairs by ignoring the indifference threshold.

Data. We use seven standard optimization benchmarks from (Surjanovic & Bingham, 2013), grouped into two sets. The first set consists of Branin, Six-Hump, and Bohachevsky, which have non-spiky landscapes and are more likely to be representative of utility functions expected in human studies. The second set consists of Levy13, Bukin6, Cross-Tray, and Ackley, which have spiky landscapes designed as challenging targets for standard optimization problems, used here to demonstrate the robustness and generality of the method. Additional results and full details are provided in the Supplement C.

The data for the real-world case study is explained in Section 4.3.

4.1 When Is CPBO Needed?

Standard PBO works track progress as a function of the number of human evaluations, which only makes sense when $c_p = 0$. To empirically study how $c_p > 0$ affects PBO performance, we evaluate three alternative variants of EUBO:

1. **Standard**: Every iteration proposes a pair (y, y') to be compared, with cost of $2c_p + c_e$.
2. **Consecutive**: Every iteration proposes one new candidate y_t, to be compared against the previous y_{t-1}, with cost of $c_p + c_e$.
3. **Multiple**: Every iteration proposes one new candidate as in the consecutive setting, which is compared against L previous candidates, with cost of $c_p + Lc_e$. We use $L = 5$ for illustration purposes.

Fig. 2 shows the simple regret as a function of the total budget B for three benchmark functions and three cost-balance scenarios: $c_p \ll c_e$, $c_p \approx c_e$, and $c_p \gg c_e$; see Supplement C.2 for details and results for other target functions and inference regret. The results exactly match the expectations:

1. When only evaluations are costly, standard PBO is optimal. However, the penalty for limiting to consecutive comparisons is small, with confidence intervals overlapping in some cases.
2. When $c_p \approx c_e$, the consecutive comparison strategy is consistently the best.
3. When production cost dominates, we observe a clear benefit from using more comparisons. Standard PBO is clearly the worst.

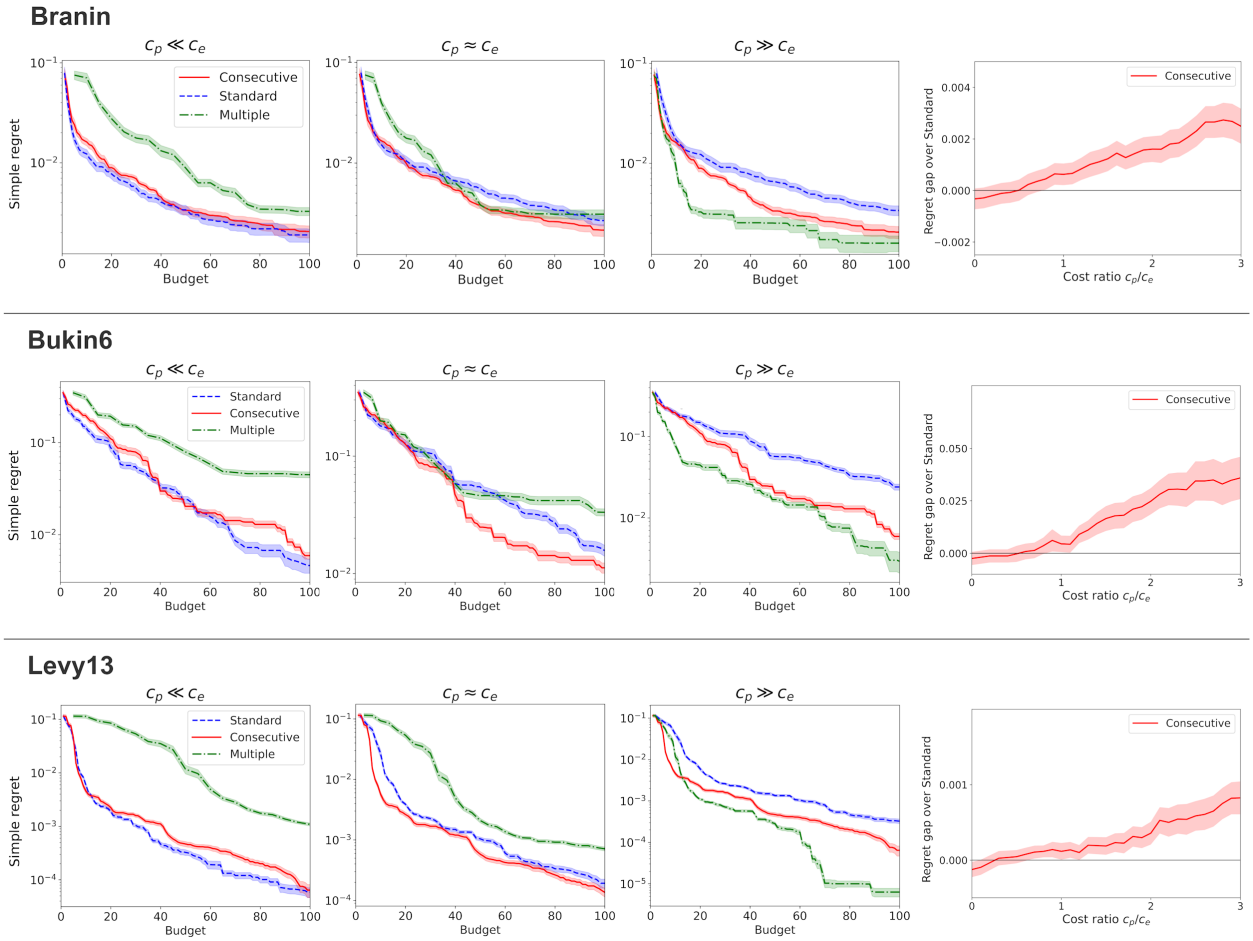

Figure 2: Convergence of PBO variants on three synthetic benchmark functions (Branin, Bukin6, and Levy13) across three cost-balance scenarios. **Left three columns:** Simple regret as a function of the total budget B when $c_p \ll c_e$, $c_p \approx c_e$, and $c_p \gg c_e$. **Right column:** Difference in simple regret between Standard and Consecutive at fixed budget $B = 100$ as the cost ratio c_p/c_e varies. Values above zero indicate that Consecutive achieves lower regret than Standard. Solid lines show the mean and shaded regions show the standard error of the mean.

That is, for extreme cases, standard PBO or a simple extension that uses all possible evaluations is the most effective option, but for balanced scenarios, we need CPBO. Fig. 2 (right most panels) quantifies the improvement from using consecutive evaluations, with roughly linear growth as a function of the cost ratio c_p/c_e.

4.2 CPBO with Just-Noticeable Differences

This section studies the overall performance of PBO methods in the CPBO setup and the importance of accounting for JND. We benchmark the methods on standard BO targets by simulating oracle responses using our user model (Eq. 4) with controlled JND threshold γ_{true}. To better understand the degree of indifference, we also report the expected ratio of indifferences over random comparison pairs for given γ_{true} for each target, denoted by $\mathbb{E}[\sim]$.

Table 1 reports the performance of the methods for $\gamma_{\text{true}} = 0.04$, corresponding to $10 - 30\%$ expected fraction of indifferences depending on the function. To ablate the importance of accounting for indifferences,

Table 1: Optimization and exploration performance measured by inference regret (Regret) and global ordinal accuracy of the utility proxy (Ord.). The results (*mean*±*sd*) are after 30 iterations, with JND of $\gamma_{\text{true}} = 0.04$. The bold font indicates the best method, with underlining indicating results that are not significantly worse (paired two-sided Wilcoxon signed-rank test, $p > 0.05$). The three functions above the dashed line best represent typical utility functions.

$f(y)$	$\mathbb{E}[\sim]$	**POP-BO**		**EUBO**		**CPBO**$_{\hat{\gamma}=0}$		**CPBO**$_{\hat{\gamma}}$	
		Regret↓	Ord.↑	Regret↓	Ord.↑	Regret↓	Ord.↑	Regret↓	Ord.↑
Branin	21%	.030 ± .024	81.5	.034 ± .032	69.1	.031 ± .019	82.0	**.020 ± .017**	**90.2**
Six-hump	34%	NA	NA	.012 ± .010	72.4	.014 ± .011	78.1	**.009 ± .004**	**84.3**
Bohachevsky	11%	NA	NA	.004 ± .004	77.2	.004 ± .005	91.7	**.001 ± .001**	**98.8**
Levy13	17%	.022 ± .015	70.5	.016 ± .012	77.0	.029 ± .031	81.9	**.011 ± .016**	**87.4**
Bukin6	10%	.182 ± .116	79.4	.126 ± .078	78.3	.155 ± .080	91.5	**.115 ± .053**	**97.0**
Cross-Tray	20%	.216 ± .093	66.9	.177 ± .112	60.2	.177 ± .107	64.1	**.150 ± .064**	**68.3**
Ackley	21%	NA	NA	.149 ± .118	72.9	.163 ± .084	79.9	**.093 ± .052**	**88.7**

we separately report the results for CPBO with the JND threshold fixed to zero (CPBO$_{\hat{\gamma}=0}$) and with a trainable threshold (CPBO$_{\hat{\gamma}}$). The former includes all other elements of our method but specifically avoids accounting for the JND.

The main result is that CPBO$_{\hat{\gamma}}$ substantially outperforms the baselines in all metrics, highlighting the importance of accounting for JND. A secondary observation is that even when not accounting for the indifferences (CPBO$_{\hat{\gamma}=0}$), we have higher ordinal accuracy compared to the more exploitative POP-BO and EUBO, with no notable difference in inference regret. This is in line with the more explorative nature of information-theoretic acquisition functions. See Supplement C.4 for a complementary evaluation metric, extended optimization curves over longer horizons and additional discussion.

An interesting question is how γ_{true}, the true degree of indifference of the oracle, influences the performance. Fig. 3 shows that when all preference comparisons are definite, the state-of-the-art EUBO is the best in terms of regret, and CPBO behaves like a standard explorative method under the binary preference model. However, already around $\gamma_{\text{true}} \approx 0.02$, when $\mathbb{E}[\sim] \approx 11\%$ for Branin and $\mathbb{E}[\sim] \approx 9\%$ for Levy13, modeling the perceptual ambiguity becomes critical. For all higher thresholds, CPBO is consistently the best, only losing its advantage when the task becomes effectively impossible (for Levy13 with $\gamma_{\text{true}} = 0.4$, we have $r_t = 0$ for 94% of comparisons).

For a broad range of $\gamma_{\text{true}} > 0$, CPBO works *better* than for $\gamma_{\text{true}} = 0$. This seemingly counter-intuitive observation has a natural explanation. The information gain Eq. (9) tends to zero for both $\Delta f_t \ll \gamma$ and $\Delta f_t \gg \gamma$ and hence the algorithm prefers candidates with $\Delta f_t \approx \gamma$. This helps prune out large parts of irrelevant configuration space and assists in proposing high-utility areas, as illustrated in Fig. 1C and other examples in the Supplement C.4.

4.3 Case Study: Extrusion Optimization

Task. We demonstrate CPBO in high-moisture extrusion processing (HMEP), a central step in a typical research process for developing new plant-based meat analogues. HMEP enables the formation of fibrous meat-like structures at an industrial scale through continuous operation, offering a sustainable alternative to conventional animal meat, which is a major contributor to greenhouse gas emissions (Dekkers et al., 2018).

A twin-screw extruder processes powdery ingredients as a continuous stream, and the properties of the extrudate (final outcome) at any given time are influenced by its settings, here $x \in \mathbb{R}^3$ indicating the temperature, water feed rate, and screw speed. The goal is to find $\hat{x}$ such that the outcome is sufficiently good for subsequent research stages, such as instrumental texture analyses and sensory evaluation by a trained panel (Nisov et al., 2024). The preliminary optimization for a new ingredient mixture may take

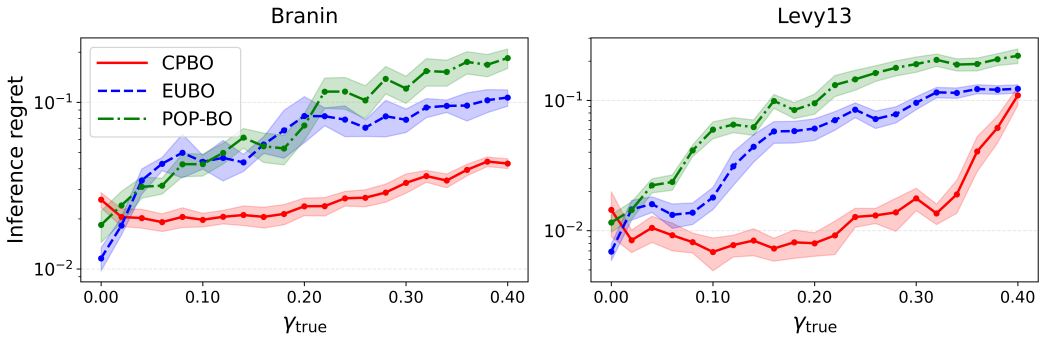

Figure 3: Inference regret on Branin and Levy13 at iteration 30 for varying degrees of JND γ_{true}. Solid line shows the mean and shaded regions the standard error of the mean.

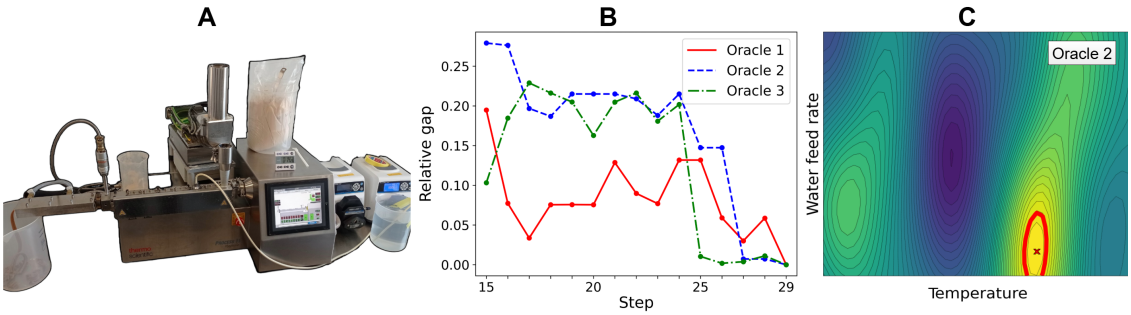

Figure 4: (A): High-moisture extruder, producing a candidate y based on current configuration x, for sensory comparison against the previous candidate. (B): Inference regret gap for three operators optimizing the configuration. (C): 2D slice of the operator's (Oracle 2) learned utility: points on or inside the red contour are predicted to be indifferent to the posterior maximizer ("x").

even a full working day and needs to be done repeatedly for alternative ingredients, making it a practical bottleneck. Moreover, the time and quality of the outcome heavily rely on the level of experience of the operator. Algorithmic support improves reproducibility across operators by reducing reliance on current subjective trial-and-error approach, especially for less experienced operators.

Extrusion Optimization as CPBO. The HMEP optimization task matches *exactly* the CPBO setup with JND. After changing the settings, we only observe the outcome once the material has propagated through the extruder. This incurs a production cost c_p, approximately 5-10 minutes per iteration. Each evaluation has a time cost c_e of the same magnitude due to the simultaneous assessment of multiple characteristics (e.g., fibrousness and gumminess). This requires a preferential comparison of overall quality rather than a numerical evaluation of the physical characteristics (Mertens et al., 2021), especially when needing to pay simultaneous attention to the extruder (Allred et al., 2016). Often the operator, especially an inexperienced one, cannot distinguish between the candidates, e.g., when both are poor in different ways, making modeling of JND critical. Finally, the candidates need to be evaluated before they cool down. This happens on time-scales comparable to the production delay, and hence only the latest candidate is suitable for reliable comparisons. Use of older samples would require re-heating, with significant retrieval cost c_r.

Experimental Setup. We conducted a study involving three operators with varying level of experience, each solving the same optimization problem by following CPBO recommendations and only indicating the subjective preferences for consecutive comparisons while operating the extruder. After 30 iterations (first 15 with a static design but still asking for preferential judgments for learning $f(x)$), we produced the candidate for $\arg\max f(x)$ as the final solution. Afterwards, each operator assessed whether the final candidate was

satisfactory, and they were given the opportunity to manually explore the configuration space to find a better solution. See Supplement D for all details.

Results. Fig. 4 shows the optimization trajectories for the three operators during the CPBO iterations. Since the true optimum is not known, we report the relative gap to the optimum at the end, corresponding to inference regret plus an unknown constant. All trajectories converge around iterations 25-29. The learned γ $(0.02, 0.04$ and $0.05)$ were consistent for the three operators, each indicating indifference 2 to 4 times; the choice of $\gamma_{\text{true}} = 0.04$ for the earlier experiment was in part motivated by the desire to make the ratio of indifferences similar to this demonstration.

All operators expressed satisfaction with the result, judging the extrudate to be suitable for subsequent stages. After carrying out 1 to 4 attempts at finding better configurations by any means possible, all three still considered the result of the optimization process the best. This does not prove it is the global optimum, but validates we met the application task need: all operators found the settings for producing good enough extrudates. Fig. 4C shows the learned utility proxy for one operator. Even though the utility landscape is multimodal, the other mode is still noticeably worse than the optimum, revealed by the JND only covering configurations around the mode. The operator confirmed this by indicating a difference between the modes.

5 Discussion and Conclusion

Despite active efforts to support AI-assisted optimization (Rainforth et al., 2024; Löfgren et al., 2020; Klami et al., 2024), complex R&D optimization tasks that do not fit the standard BO setup remain unsupported in practice. Our main arguments and contributions relating to better accounting for the costs and operational constraints may seem obvious in hindsight, also because of the massive empirical improvement obtained with relatively simple modifications. However, the current PBO literature indeed ignores both the cost of producing the candidates and the obvious need for experts to indicate indifference. We speculate this is in part due to the community focusing on clean benchmarks and computational BO tasks, such as hyperparameter optimization (Bergstra et al., 2013; Snoek et al., 2012), often neglecting empirical sciences that may have the biggest need for these tools.

Our main contribution is the task formulation and the identification of its necessary components, rather than the specific algorithm. However, the proposed method is already highly robust and easy to use in practice, not requiring any adaptation or hyperparameter tuning for the different target functions (including the real use case). It could, however, be made more computationally efficient, for example by amortizing the information gain computation (Volpp et al., 2020; Swersky et al., 2020). Empirical validation could also be extended to higher dimensions. Here we considered only problems of at most six dimensions (see Supplement C.4) to align with the typical use-cases in empirical science. For higher dimensionalities, CPBO should scale like any PBO method relying on pairwise comparison.

We demonstrated CPBO in empirical research in food science, aiming to speed-up food extrusion processing. However, the range of potential applications is much broader: any problem with non-zero candidate production cost readily leads to a consecutive optimization formulation, and JND is unlikely to be zero in any real-world setting. For example, limitations of human visual, auditory, and tactile perception alone make it impossible to distinguish tiny differences in stimuli (Johnson et al., 2002), which makes this work directly relevant to various human-in-the-loop formulations (Catkin & Patoglu, 2023; Adachi et al., 2024).

References

Masaki Adachi, Brady Planden, David Howey, Michael A. Osborne, Sebastian Orbell, Natalia Ares, Krikamol Muandet, and Siu Lun Chau. Looping in the human: Collaborative and explainable Bayesian optimization. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, pp. 505–513, 2024.

Arpit Agarwal, Rohan Ghuge, and Viswanath Nagarajan. Batched dueling bandits. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 89–110, 2022.

Sarah R. Allred, L. Elizabeth Crawford, Sean Duffy, and John Smith. Working memory and spatial judgments: Cognitive load increases the central tendency bias. *Psychonomic Bulletin & Review*, 23(6):1825–1831, 2016.

Raul Astudillo, Zhiyuan Jerry Lin, Eytan Bakshy, and Peter Frazier. qEUBO: A decision-theoretic acquisition function for preferential Bayesian optimization. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, pp. 1093–1114, 2023.

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.

Alessio Benavoli and Dario Azzimonti. A tutorial on learning from preferences and choices with gaussian processes, 2024.

James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 115–123, 2013.

Ralph Allan Bradley and Milton Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Eric Brochu, Tyson Brochu, and Nando De Freitas. A bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 103–112, 2010.

Bilal Catkin and Volkan Patoglu. Preference-based human-in-the-loop optimization for perceived realism of haptic rendering. *IEEE Transactions on Haptics*, 16(4):470–476, 2023.

Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 137–144, 2005.

Roger R. Davidson. On extending the Bradley–Terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65(329):317–328, 1970.

Birgit L. Dekkers, Remko M. Boom, and Atze Jan van der Goot. Structuring processes for meat analogues. *Trends in Food Science & Technology*, 81:25–36, 2018.

Jacob Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Javier González, Zhenwen Dai, Andreas Damianou, and Neil D. Lawrence. Preferential Bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1282–1291, 2017.

Kenneth O. Johnson, Steven S. Hsiao, and Takashi Yoshioka. Neural coding and the basic law of psychophysics. *The Neuroscientist*, 8(2):111–121, 2002.

Arto Klami, Theo Damoulas, Ola Engkvist, Patrick Rinke, and Samuel Kaski. Virtual laboratories: Transforming research with AI. *Data-Centric Engineering*, 5:e19, 2024.

Greg Kochanski, Daniel Golovin, John Karro, Benjamin Solnik, Subhodeep Moitra, and D. Sculley. Bayesian optimization for a better dessert. In *Proceedings of the NeurIPS 2017 Workshop on Bayesian Optimization*, 2017.

Harry T. Lawless and Hildegarde Heymann. *Sensory Evaluation of Food: Principles and Practices*. Springer Science & Business Media, 2010.

Eric Hans Lee, Valerio Perrone, Cedric Archambeau, and Matthias Seeger. Cost-aware Bayesian optimization. arXiv preprint arXiv:2003.10870, 2020.

Zhiyuan Jerry Lin, Raul Astudillo, Peter I. Frazier, and Eytan Bakshy. Preference exploration for efficient Bayesian optimization with multiple outcomes. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, 2022.

Jinsong Liu, Dongdong Ge, and Ruihao Zhu. Reward learning from preference with ties. arXiv preprint arXiv:2410.05328, 2024.

Joakim Löfgren, Milica Todorovic, Patrick Rinke, Henri Paulamäki, Arttu Tolvanen, Ville Parkkinen, Ulpu Remes, and Nuutti Sten. Bayesian optimization structure search (BOSS), 2020.

R. Duncan Luce. *Individual Choice Behavior*. John Wiley, 1959.

Phuc Luong, Dang Nguyen, Sunil Gupta, Santu Rana, and Svetha Venkatesh. Adaptive cost-aware Bayesian optimization. *Knowledge-Based Systems*, 232:107481, 2021.

Charles F. Manski. The structure of random utility models. *Theory and Decision*, 8(3):229–254, 1977.

Daniel McFadden. Conditional logit analysis of qualitative choice behavior. In *Frontiers in Econometrics*, pp. 105–142. Academic Press, New York, 1974.

M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

Alica Mertens, Ulf K. Mertens, and Veronika Lerche. On the difficulty to think in ratios: A methodological bias in stevens' magnitude estimation procedure. *Attention, Perception, & Psychophysics*, 83(5):2347–2365, 2021.

Petrus Mikkola, Milica Todorović, Jari Järvi, Patrick Rinke, and Samuel Kaski. Projective preferential Bayesian optimization. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 6884–6892, 2020.

Frederick Mosteller. Remarks on the method of paired comparisons: I. the least squares solution assuming equal standard deviations and equal correlations. *Psychometrika*, 16(1):3–9, 1951.

Quoc Phong Nguyen, Sebastian Tay, Bryan Kian Hsiang Low, and Patrick Jaillet. Top-k ranking Bayesian optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:9135–9143, 2021.

Anni Nisov, Anniina Valtonen, Heikki Aisala, Andrea Spaccasassi, Christoph Walser, Corinna Dawid, and Nesli Sozer. Effect of peptide formation during rapeseed fermentation on meat analogue structure and sensory properties at different pH conditions. *Food Research International*, 180:114070, 2024.

Robin L Plackett. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202, 1975.

Tom Rainforth, Adam Foster, Desi R. Ivanova, and Freddie Bickford Smith. Modern Bayesian experimental design. *Statistical Science*, 39(1):100–114, 2024.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. `https://www.sfu.ca/~ssurjano/`, 2013. Accessed: 2026-02-11.

Kevin Swersky, Jasper Snoek, and Ryan P. Adams. Multi-task Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Kevin Swersky, Yulia Rubanova, David Dohan, and Kevin Murphy. Amortized Bayesian optimization over discrete spaces. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence*, pp. 769–778, 2020.

L. L. Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273–286, 1927.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, pp. 567–574, 2009.

Michael Volpp, Lukas P. Fröhlich, Kirsten Fischer, Andreas Doerr, Stefan Falkner, Frank Hutter, and Christian Daniel. Meta-learning acquisition functions for transfer learning in Bayesian optimization. In *Proceedings of the International Conference on Learning Representations*, 2020.

Zi Wang and Stefanie Jegelka. Max-Value entropy search for efficient Bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3627–3635, 2017.

Ernst Heinrich Weber. The sense of Touch. In H. E. Ross and D. J. Murray (eds.), *The Sense of Touch*. Academic Press, New York, NY, 1978.

Christopher K. I. Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Wenjie Xu, Wenbin Wang, Yuning Jiang, Bratislav Svetozarevic, and Colin N. Jones. Principled preferential Bayesian optimization. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Ke Zhou, Gui-Rong Xue, Hongyuan Zha, and Yong Yu. Learning to rank with ties. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 275–282, 2008.

A Appendix

The appendix complements the main manuscript, providing additional method details (Section B), details of the empirical experiments on the benchmark datasets with additional illustrations and numerical results (Section C), and description of the high-moisture extrusion experiment (Section D). Code implementing the algorithm and reproducing the synthetic data experiments is provided within the zip file.

All figures and tables in this supplement are numbered separately from the main paper, using the format Figure S1, Table S1, and similar.

B Method Details

In this section, we describe the implementation and hyperparameter choices used in our experiments. First, we discuss the surrogate GP model and the three-outcome likelihood with JND, and later the details about the acquisition function.

B.1 Surrogate GP Model and Likelihood

To remove the scale ambiguity in Eq. 1, we fix $\sigma_f^2 = 10.0$. Inference is performed using a variational approximation to the posterior $p(f \mid \mathcal{D}_t)$ over the latent utility function $f(x)$. At iteration t, we define the variational distribution over the latent utility values $f(x_i)$ corresponding to all observed configurations $x_i{}_{i=1}^{t}$. The corresponding latent utility values are denoted as $\mathbf{u} = [f(x_1), \ldots, f(x_t)]^\top$.

A diagonal jitter of 10^{-4} is added to all kernel covariance matrices to ensure numerical stability during Cholesky decompositions. We implement the model using GPyTorch's variational GP framework (Gardner et al., 2018), specifically the `ApproximateGP` class with a Cholesky-parameterized variational distribution and a learned inducing set. While we use all training points as inducing inputs in our experiments (effectively removing the sparsity), our model follows the variational GP framework introduced by Titsias (2009), as implemented in GPyTorch's `VariationalStrategy`.

While we do not apply any prior on the ARD lengthscales $\{\ell_d\}$ in the benchmark experiments, we found it occasionally helpful in practice to encourage moderate smoothness early in training. In particular, for the real-world extrusion experiment, we placed independent Gamma priors $\ell_d \sim \text{Gamma}(1.0, 0.05)$ (shape, rate) on each lengthscale dimension.

Likelihood Implementation. The expected log-likelihood in the ELBO is approximated by averaging over $S = 50$ samples $f^{(s)}$ drawn from the variational posterior $q(f)$

$$\mathbb{E}_{q(f)}[\log P(\mathcal{D}_t \mid f)] \approx \frac{1}{S} \sum_{s=1}^{S} \log P(\mathcal{D}_t \mid f^{(s)}). \tag{S1}$$

We found $S = 50$ to be sufficiently accurate in practice and did not explore smaller values, as the overhead was acceptable relative to overall runtime. To prevent numerical issues, we add a small constant $\varepsilon = 10^{-5}$ to the computed probabilities before applying the logarithm.

The perceptual noise parameter is fixed to $\sigma = 0.04$ to eliminate scale ambiguity between utility magnitude and noise, and sensitivity to this choice is evaluated in Section C.4.

Optimization. We jointly optimize:

- the variational parameters $\mathbf{m}_u$ and $\mathbf{L}$,
- the ARD lengthscales $\{\ell_d\}$ and the prior mean constant m,
- and the JND threshold γ.

Optimization is performed by maximizing the ELBO using full-batch Adam (learning rate 10^{-2}) for a fixed number of 2000 iterations, which we found sufficient for the approximation to converge in all experiments. During training, we retain the model parameters that achieve the lowest loss across all iterations and return them for evaluation.

B.2 Acquisition Implementation Details

We describe key implementation choices that improve the numerical stability and computational efficiency of our acquisition function based on MES adapted for CPBO.

Zero-Mean Normalization of Samples. To ensure numerical stability and comparability during acquisition, we apply zero-mean normalization to every posterior sample $f^{(s)}$ drawn when computing entropy terms or estimating the Gumbel distribution. Given a batch $\mathcal{X}_{\text{eval}} = \{x^{(1)}, \ldots, x^{(N)}\}$, we define:

$$\tilde{f}^{(s)}(x^{(i)}) = f^{(s)}(x^{(i)}) - \frac{1}{N}\sum_{j=1}^{N} f^{(s)}(x^{(j)}), \quad \text{for } i = 1, \ldots, N. \tag{S2}$$

This transformation preserves preference structure (based on utility differences) while eliminating arbitrary shifts in the sample-wise utility scale. It improves stability and alignment across Monte Carlo samples used in acquisition.

Unconditional Entropy. We estimate $H(R_{t+1} \mid \mathcal{D}_t)$, using 1000 Monte Carlo samples from the zero-centered GP posterior. Predictive probabilities are computed from the three-outcome likelihood with fixed σ and learned γ. As above, a small constant $\varepsilon = 10^{-5}$ is added to all probabilities before taking logarithms for numerical stability.

Conditional Entropy Approximation. The main computational bottleneck in our acquisition is evaluating the expectation over utility maxima $f^* \sim p(f^*)$ in the conditional entropy term. A naive solution would require drawing many f^* samples and computing the full conditional preference distribution for each, which is prohibitively expensive due to the need for repeated truncated sampling. Instead, we use an efficient binning strategy

- We first sample 25,000 utility maxima f^* from the Gumbel distribution G fitted to the GP posterior. This relatively large number was chosen to improve stability and to align computation time with the natural pause between consecutive runs in our real extrusion experiment.
- The samples are grouped into 20 equally spaced bins, and the mean value $\bar{f}^*$ within each bin is used as a representative.

This reduces the number of truncation conditions to just 20. For each $\bar{f}^*$, we draw 1000 truncated utility samples $f^{(s)}(\cdot) \sim p(f \mid \max f(\cdot) \leq \bar{f}^*)$, apply zero-centering, and compute conditional preference distributions accordingly.

Gumbel Fitting and Sampling. We approximate the distribution of utility maxima f^* by fitting a Gumbel distribution to posterior samples from the GP, following the approach of Wang & Jegelka (2017). Specifically, we estimate the 25th, 50th, and 75th percentiles (y_1, y_2, y_3) of the utility samples corresponding to ranks $r_1 = 0.25$, $r_2 = 0.75$, and $r_3 = 0.50$, and compute the scale and location parameters as

$$b = \frac{y_1 - y_2}{\log(-\log r_2) - \log(-\log r_1)}, \quad a = y_3 + b \cdot \log(-\log r_3). \tag{S3}$$

While the original MES formulation uses only the 25th and 75th percentiles, we also include the median percentile ($r_3 = 0.50$) to better center the fit, which we found to yield slightly improved empirical accuracy and stability. This choice also aligns with the convention used in the existing implementation of MES in BoTorch. To improve numerical stability and reduce computation time, we restrict Gumbel rank sampling to the interval $[\varepsilon_{\text{r}}, 1 - \varepsilon_{\text{r}}]$ rather than the full $[0, 1]$, which avoids extreme tail values that are difficult to truncate and slow to evaluate. We use $\varepsilon_{\text{r}} = 0.01$ in all experiments.

Acquisition Maximization. Rather than evaluating the information gain acquisition function $\alpha(x)$ exhaustively over $x \in \mathcal{X}$, we optimize it using a global black-box optimization routine to find informative candidates. Specifically, we use Bayesian optimization as the optimizer, since it does not require gradient information and can handle noisy function evaluations resulting from the conditional entropy approximation. Let $\hat{x} = \arg\max_{x \in \mathcal{X}} \alpha(x)$. We approximate $\hat{x}$ by running an internal Bayesian optimization loop using GPyOpt, which employs a Gaussian process surrogate with a Matérn 5/2 kernel and ARD. The procedure starts with 10 random initial evaluations, followed by 25 optimization steps using expected improvement as the acquisition strategy. This procedure yields a computationally efficient approximation of $\hat{x}$ without resorting to full grid evaluation.

C Extended Experiments And Details

This section provides full experimental details and additional results supporting the main paper. We organize the content into five parts:

1. **Benchmark functions:** Additional details on the benchmark functions used in the experiments.
2. **Details for "When is CPBO needed?":** Detailed explanation of the production and evaluation cost modeling used, including cost budgets and behavior under different regimes.
3. **Baselines and Adaptation to CPBO:** Description of how EUBO and POP-BO were adapted to fit the CPBO setting.
4. **Additional CPBO benchmarks:** Additional CPBO benchmarks, reporting results for all seven standard 2D functions under varying JND thresholds γ_{true}. This includes the utility recovery and choice accuracy metric analysis, ablations on the perceptual-noise scale σ, higher-dimensional benchmarks (Alpine1 6D and Levy 6D), runtime scaling with dimensionality, and a supplementary comparison against qEUBO.
5. **Additional visualizations:** Complementary plots such as learned JND threshold, latent utility surfaces, and JND band across different functions.

Synthetic experiments were run on a computing cluster provided by an external party (name removed for double-blind submission), configured to use 4 CPU cores and 10 GB RAM per job. Real-world optimization was performed on a laptop with an Intel i7-9750H processor and 16 GB RAM. Acquisition parameters that impact runtime were configured so each BO step completed in approximately 5 minutes, aligning with the 5-10 minute interval required between real-life extrusion runs.

C.1 Benchmark Functions

We consider a set of standard continuous benchmark functions (Surjanovic & Bingham, 2013) commonly used in the Bayesian optimization literature. For readability in tables and figures, the *Six-Hump Camelback* function is referred to as **Six-hump**, and the *Cross-in-Tray* function as **Cross-Tray**.

All benchmark functions are normalized to the $[0, 1]$ range to ensure consistent interpretation of the JND thresholds γ_{true} and to allow fair comparison across different functions. An overview of the normalized utility surfaces is provided in Figure S5, which illustrates the function landscapes and their corresponding indifference regions.

C.2 Details for 'When Is CPBO Needed?'

Section 4.1 introduced three preferential BO strategies that differ in how candidates are generated and compared, under varying assumptions about production and evaluation cost. Here, we provide additional results to support that analysis.

The cost regimes used in our experiments are summarized in Table S1, covering settings where either production, evaluation, or both are costly.

Table S1: Cost regimes used in our experiments

Regime	**Production cost** c_p	**Evaluation cost** c_e
$c_p \ll c_e$	0	1
$c_p \approx c_e$	1	1
$c_p \gg c_e$	1	0

Each strategy incurs a different per-iteration cost that depends on the relative weights of production and evaluation, denoted by w_p and w_e. The total cost per iteration is computed as:

$$\text{Iteration cost} = c_p \cdot w_p + c_e \cdot w_e. \tag{S4}$$

The weighting terms (w_p, w_e) vary by comparison strategy as follows:

- **Standard:** Compares two new candidates; $w_p = 2$, $w_e = 1$.
- **Consecutive:** Compares the current candidate to the previous one; $w_p = 1$, $w_e = 1$.
- **Multiple:** Uses the same acquisition strategy as the consecutive case (conditioning on the previous candidate) but compares the new candidate against $L = 5$ previously produced candidates, including the latest; $w_p = 1$, $w_e = 5$.

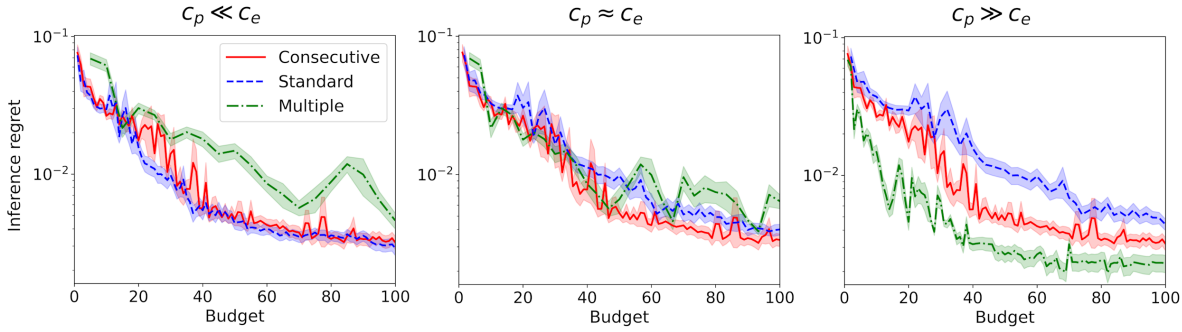

Figure S1: Convergence of PBO variants on Branin, for three cost-balance scenarios.

To complement the simple regret results shown in Figure 2 in the main paper, Figure S1 shows inference regret for Branin under a fixed budget of 100 units. Tables S2 and S3 provide corresponding numerical results under a fixed cost budget of 30 units. The smaller budget is used in tables for readability, as regret values become very small at 100 units.

Table S2: Simple regret under a fixed budget of 30 cost-units ($B = 30$). Values are mean ± one standard deviation. The bold font indicates the best method within each function and cost scenario, with underlining indicating results that are not significantly worse (paired two-sided Wilcoxon signed-rank test, $p > 0.05$).

$f(y)$	$c_p \ll c_e$			$c_p \approx c_e$			$c_p \gg c_e$		
	Standard	**Consecutive**	**Multiple**	**Standard**	**Consecutive**	**Multiple**	**Standard**	**Consecutive**	**Multiple**
Branin	**.006 ± .003**	.007 ± .003	.015 ± .012	.012 ± .008	**.011 ± .007**	.019 ± .015	.009 ± .006	.007 ± .003	**.003 ± .002**
Six-hump	**.0004 ± .0003**	.0007 ± .0008	.096 ± .079	.0014 ± .0011	**.0009 ± .0005**	.0110 ± .0091	.0021 ± .0019	.0015 ± .0012	**.0002 ± .0001**
Bohachevsky	**.0001 ± .0002**	.002 ± .004	.086 ± .082	.005 ± .006	**.003 ± .003**	.131 ± .106	.002 ± .003	.002 ± .003	**.0001 ± .001**
Levy13	**.001 ± .001**	.014 ± .043	.051 ± .046	**.006 ± .004**	.007 ± .005	.061 ± .054	.003 ± .001	.014 ± .043	**.001 ± .001**
Bukin6	**.053 ± .042**	.078 ± .076	.148 ± .070	.170 ± .091	**.137 ± .082**	.156 ± .076	.107 ± .092	.067 ± .066	**.033 ± .023**
Cross-Tray	.040 ± .043	**.037 ± .035**	.131 ± .078	.084 ± .079	**.074 ± .052**	.134 ± .082	.051 ± .067	.038 ± .037	**.029 ± .035**
Ackley	**.017 ± .041**	.040 ± .036	.227 ± .103	.112 ± .089	**.112 ± .054**	.248 ± .099	.086 ± .093	.039 ± .034	**.005 ± .004**

Table S3: Inference regret under a fixed budget of 30 cost-units ($B = 30$). Values are mean ± one standard deviation. The bold font indicates the best method within each function and cost scenario, with underlining indicating results that are not significantly worse (paired two-sided Wilcoxon signed-rank test, $p > 0.05$).

$f(y)$	$c_p \ll c_e$			$c_p \approx c_e$			$c_p \gg c_e$		
	Standard	**Consecutive**	**Multiple**	**Standard**	**Consecutive**	**Multiple**	**Standard**	**Consecutive**	**Multiple**
Branin	**.009 ± .008**	.014 ± .015	.018 ± .013	.028 ± .015	**.023 ± .017**	.026 ± .015	.024 ± .042	.015 ± .016	**.005 ± .005**
Six-hump	**.0016 ± .0015**	.0025 ± .0023	.0139 ± .0114	.0038 ± .0032	**.0028 ± .0020**	.0143 ± .0127	.0044 ± .0040	.0032 ± .0025	**.0009 ± .0006**
Bohachevsky	**.002 ± .004**	.141 ± .135	.077 ± .089	.054 ± .067	**.015 ± .012**	.109 ± .101	.117 ± .119	.172 ± .139	**.001 ± .001**
Levy13	**.002 ± .002**	.071 ± .142	.044 ± .052	**.009 ± .007**	<u>.011 ± .008</u>	.052 ± .051	.061 ± .086	.056 ± .075	**.024 ± .051**
Bukin6	**.091 ± .044**	.232 ± .097	.192 ± .084	.282 ± .155	**.251 ± .118**	.262 ± .114	.235 ± .115	.194 ± .110	**.084 ± .029**
Cross-Tray	**.131 ± .097**	<u>.142 ± .105</u>	.196 ± .131	<u>.161 ± .128</u>	**.156 ± .076**	.183 ± .080	.188 ± .102	.155 ± .110	**.124 ± .107**
Ackley	**.047 ± .044**	.152 ± .129	.203 ± .122	.163 ± .091	**.132 ± .058**	.235 ± .102	.192 ± .141	.153 ± .131	**.136 ± .114**

C.3 Baselines and Adaptation to CPBO

Here we explain how the comparison methods, EUBO and POP-BO, were used in the experiment reported in Section 4.2.

Oracle model. All baselines are trained using a common simulated oracle based on the three-outcome model, but adapted to produce binary labels. Specifically, if the noisy utility difference falls within the JND threshold γ_{true}, the oracle returns a random label in $\{-1, +1\}$:

$$r_t = \begin{cases} +1 & \text{if } \Delta f_t + \delta_t > \gamma_{\text{true}}, \\ -1 & \text{if } \Delta f_t + \delta_t < -\gamma_{\text{true}}, \\ \sim \text{Uniform}(\{-1, +1\}) & \text{otherwise.} \end{cases} \tag{S5}$$

This ensures compatibility with baseline methods that assume binary feedback, while still capturing the ambiguity near the threshold.

EUBO Adaptation. We use the official `BoTorch` implementation of EUBO, modifying it to enforce the CPBO constraint of consecutive comparisons. Technically, we propose a single new candidate per iteration, and we condition the query on the candidate selected in the previous iteration. This restricts each query to compare the new candidate against the one produced in the prior step.

POP-BO Setup. POP-BO is inherently designed to operate with consecutive preference comparisons and requires no structural adaptation for the CPBO setting. However, its original implementation uses a binary preference oracle based on the Bradley-Terry model, where the probability of one configuration being preferred is given by $\sigma(f(x) - f(x'))$, with $\sigma(\cdot)$ denoting the logistic function. To ensure consistency across methods, we replace this with the Thurstone-style oracle described above.

In their original implementation, POP-BO assumes access to ground-truth function values in order to manually set kernel hyperparameters on a per-function basis. This is not applicable in real-world preference-based settings, where utility values are unobserved. As the original paper does not provide a method for estimating these parameters from preferences, and our own attempts proved unstable, we instead follow their published configuration and fix the hyperparameters for each benchmark.

C.4 CPBO Benchmarks

We extend the main paper results by evaluating CPBO under a broader range of indifference levels to examine its robustness and generality. Specifically, we test how performance varies as the true JND threshold γ_{true} and the expected proportion of indifferences $\mathbb{E}[\sim]$ change across benchmark functions.

We consider two complementary setups:

- Fixed threshold $\gamma_{\text{true}} = 0.1$, corresponding to approximately 25–50% expected indifferences $\mathbb{E}[\sim]$.
- Fixed $\mathbb{E}[\sim]$ values of 15% and 30%, with per-function γ_{true} chosen accordingly.

Tables S4 to S6 report inference regret and ordinal accuracy across these settings. CPBO with learned $\hat{\gamma}$ achieves the strongest overall performance and remains competitive across all benchmarks, including against a variant with fixed $\hat{\gamma} = 0$, which highlights the benefit of explicitly modeling perceptual ambiguity. Overall, CPBO demonstrates stable optimization, accurate utility recovery, and robustness across a wide range of $\mathbb{E}[\sim]$ levels.

Table S4: Optimization and exploration performance measured by inference regret (Regret) and global ordinal accuracy of the utility proxy (Ord.). Results (mean ±sd) after 30 iterations under $\gamma_{\text{true}} = 0.1$. The bold font indicates the best method, with underlining indicating results that are not significantly worse (paired two-sided Wilcoxon signed-rank test, $p > 0.05$).

$f(y)$	$\mathbb{E}[\sim]$	**POP-BO**		**EUBO**		**CPBO**$_{\hat{\gamma}=0}$		**CPBO**$_{\hat{\gamma}}$	
		Regret↓	Ord.↑	Regret↓	Ord.↑	Regret↓	Ord.↑	Regret↓	Ord.↑
Branin	43%	.042 ± .035	75.8	.044 ± .023	68.2	.046 ± .016	77.9	**.021 ± .017**	**89.5**
Six-hump	54%	NA	NA	.029 ± .017	68.2	.035 ± .022	73.1	**.012 ± .007**	**83.1**
Bohachevsky	25%	NA	NA	.021 ± .018	73.6	.024 ± .041	89.2	**.001 ± .002**	**98.6**
Levy13	39%	.059 ± .044	67.1	.018 ± .019	75.7	.056 ± .057	80.1	**.007 ± .007**	**87.4**
Bukin6	25%	.204 ± .086	77.9	**.182 ± .110**	76.4	.266 ± .105	88.3	.184 ± .101	**95.3**
Cross-Tray	47%	.245 ± .122	63.8	.198 ± .129	60.7	.218 ± .104	64.0	**.152 ± .074**	**69.7**
Ackley	48%	NA	NA	.157 ± .117	72.8	.238 ± .128	77.3	**.122 ± .059**	**87.9**

Table S5: Optimization and exploration performance measured by inference regret (Regret) and global ordinal accuracy of the utility proxy (Ord.). Results (mean ±sd) after 30 iterations with $\mathbb{E}[\sim] = 30\%$. The bold font indicates the best method, with underlining indicating results that are not significantly worse (paired two-sided Wilcoxon signed-rank test, $p > 0.05$).

$f(y)$	γ_{true}	**POP-BO**		**EUBO**		**CPBO**$_{\hat{\gamma}=0}$		**CPBO**$_{\hat{\gamma}}$	
		Regret↓	Ord.↑	Regret↓	Ord.↑	Regret↓	Ord.↑	Regret↓	Ord.↑
Branin	.061	.031 ± .018	77.3	.042 ± .020	67.8	.033 ± .024	80.4	**.020 ± .014**	**90.3**
Six-hump	.033	NA	NA	.013 ± .009	73.0	.014 ± .010	78.0	**.010 ± .004**	**84.0**
Bohachevsky	.120	NA	NA	.022 ± .018	74.0	.028 ± .033	88.6	**.001 ± .002**	**97.9**
Levy13	.074	.035 ± .022	69.1	.017 ± .012	74.9	.046 ± .053	80.4	**.010 ± .008**	**88.6**
Bukin6	.122	.208 ± .103	76.4	.202 ± .103	77.1	.272 ± .104	87.7	**.188 ± .092**	**94.7**
Cross-Tray	.062	.219 ± .103	66.1	.191 ± .124	61.1	.194 ± .096	63.9	**.138 ± .071**	**67.7**
Ackley	.059	NA	NA	.158 ± .107	75.0	.174 ± .072	77.9	**.110 ± .055**	**89.1**

Table S6: Optimization and exploration performance measured by inference regret (Regret) and global ordinal accuracy of the utility proxy (Ord.). Results (mean ±sd) after 30 iterations with $\mathbb{E}[\sim] = 15\%$. The bold font indicates the best method, with underlining indicating results that are not significantly worse (paired two-sided Wilcoxon signed-rank test, $p > 0.05$).

$f(y)$	γ_{true}	**POP-BO**		**EUBO**		**CPBO**$_{\hat{\gamma}=0}$		**CPBO**$_{\hat{\gamma}}$	
		Regret↓	Ord.↑	Regret↓	Ord.↑	Regret↓	Ord.↑	Regret↓	Ord.↑
Branin	.0279	.025 ± .019	81.3	**.018 ± .013**	70.2	.026 ± .017	81.2	.020 ± .017	**90.1**
Six-hump	.0103	NA	NA	**.006 ± .004**	73.2	.012 ± .008	78.3	.008 ± .004	**86.2**
Bohachevsky	.0578	NA	NA	.011 ± .012	77.5	.009 ± .018	90.6	**.001 ± .001**	**98.7**
Levy13	.0355	.016 ± .012	70.9	.015 ± .011	75.8	.028 ± .029	82.5	**.008 ± .009**	**87.1**
Bukin6	.0596	.195 ± .115	78.7	.171 ± .114	75.3	.166 ± .080	90.1	**.159 ± .085**	**96.3**
Cross-Tray	.0302	.196 ± .088	67.6	.160 ± .091	63.1	.166 ± .081	64.3	**.153 ± .076**	**68.3**
Ackley	.0287	NA	NA	.122 ± .058	78.1	.126 ± .068	81.2	**.106 ± .047**	**85.8**

Utility Recovery and Choice Accuracy. In addition to ordinal accuracy, we also propose a *choice accuracy* metric for evaluating the exploration performance of the PBO method. It measures how often the model correctly predicts the outcome of a uniformly random pairwise comparison. Each comparison can have three possible outcomes: better, worse, or indifferent, depending on whether the true utility difference exceeds the JND threshold γ_{true}. While ordinal accuracy ignores indifference and only assesses the relative ranking between configurations, choice accuracy evaluates all three outcomes by incorporating the learned JND threshold $\hat{\gamma}$.

For baseline models that do not account for perceptual uncertainty, predictions falling within the true JND band are always treated as incorrect, since such models cannot represent indifference. Table S7 reports both ordinal and choice accuracy under $\gamma_{\text{true}} = 0.04$, complementing Table 1 in the main paper by including the corresponding standard deviations. Together, these results show how accurately each method recovers the underlying utility and predicts human-like preference behavior.

Table S7: Latent-space prediction ordinal (Ord.) and choice accuracy (%) at iteration 30 under $\gamma_{\text{true}} = 0.04$, reported as mean ± sd. The bold font indicates the best method, with underlining indicating results that are not significantly worse (paired two-sided Wilcoxon signed-rank test, $p > 0.05$).

$f(y)$	**POP-BO**		**EUBO**		**CPBO**$_{\hat{\gamma}=0}$		**CPBO**$_{\hat{\gamma}}$	
	Ord.	**Choice**	**Ord.**	**Choice**	**Ord.**	**Choice**	**Ord.**	**Choice**
Branin	81.5 ± 4.2	72.2 ± 4.3	69.1 ± 6.3	58.9 ± 6.3	82.0 ± 3.6	71.3 ± 3.7	**90.2 ± 2.3**	**77.9 ± 2.1**
Six-hump	NA	NA	72.4 ± 5.9	63.8 ± 6.0	78.1 ± 3.8	68.7 ± 3.9	**84.3 ± 2.5**	**73.8 ± 2.7**
Bohachevsky	NA	NA	77.2 ± 7.3	69.5 ± 6.8	91.7 ± 2.1	83.2 ± 2.5	**98.8 ± 1.0**	**91.5 ± 1.0**
Levy13	70.5 ± 4.1	64.3 ± 4.1	77.0 ± 6.1	66.6 ± 6.3	81.9 ± 3.7	70.3 ± 3.9	**87.4 ± 2.5**	**76.9 ± 2.2**
Bukin6	79.4 ± 4.5	74.8 ± 4.4	78.3 ± 5.1	70.9 ± 5.2	91.5 ± 2.0	82.8 ± 2.3	**97.0 ± 1.0**	**90.0 ± 1.0**
Cross-Tray	66.9 ± 3.4	**60.0 ± 3.5**	60.2 ± 5.9	50.1 ± 6.1	64.1 ± 6.9	53.9 ± 6.8	**68.3 ± 6.4**	56.6 ± 6.5
Ackley	NA	NA	72.9 ± 8.2	62.4 ± 7.6	79.9 ± 3.4	70.3 ± 3.8	**88.7 ± 2.4**	**74.9 ± 2.0**

Higher-Dimensional Benchmarks (6D and 12D). All synthetic experiments in the main paper were for 2-dimensional problems, simply for ease of visualization. The method itself naturally extends to higher dimensionalities. Figure S2 shows this in a problem of moderate dimensionality, using two 6-dimensional standard optimization functions: Alpine1 6D and Levy 6D, in comparison to EUBO. All experimental protocols, including the oracle model, CPBO configuration, and EUBO adaptation (run in a consecutive comparison mode), follow the same setup as in the 2D benchmark experiments. The utility values are

Table S8: Inference regret (mean ± sd) on Alpine1 in 12D for increasing optimization horizons.

Method	30 steps	50 steps	75 steps	100 steps
CPBO	**.158 ± .081**	**.077 ± .053**	**.047 ± .039**	**.022 ± .019**
EUBO	.262 ± .076	.251 ± .075	.236 ± .068	.214 ± .060

normalized to $[0, 1]$, and the perceptual noise used in the oracle is fixed to $\gamma_{\text{true}} = 0.04$. The results are consistent with those reported for the 2D experiments in the main paper, with CPBO clearly outperforming the comparison methods.

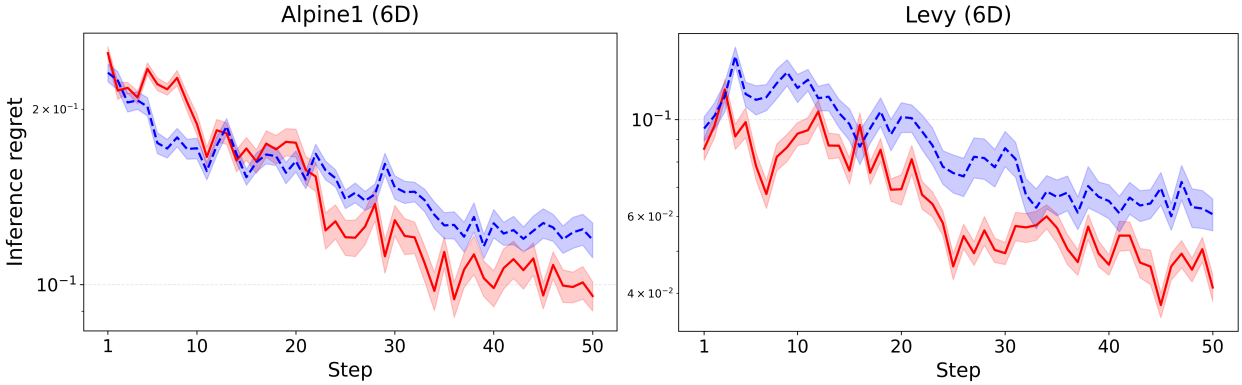

Figure S2: *Inference regret* as a function of steps under $\gamma_{\text{true}} = 0.04$. (Left) 6D Alpine1 function; (Right) 6D Levy function. Solid line shows the mean and shaded regions the standard error of the mean.

To further assess scalability, we additionally consider a higher-dimensional Alpine1 benchmark in 12D and extend the evaluation horizon up to 100 iterations. The experimental setup is identical to the 2D benchmarks, with the only difference being a slightly increased computational budget for acquisition optimization (35 optimization steps instead of 25) to account for the higher-dimensional search space. The results, reported in Table S8, show that the advantage of CPBO persists both in higher dimensions and over longer optimization horizons.

Scaling with Dimensionality. The computational complexity of CPBO scales with dimensionality in the same way as the original MES, which has been demonstrated to extend to tens of dimensions. For practical human-in-the-loop scenarios, wall-clock time per iteration is the relevant measure. In the real-world extrusion experiment (Section 4.3), our configuration resulted in an average of about five minutes per iteration, including visualization and plotting overhead, which conveniently matches the time required to produce a new sample with the machine.

To illustrate scalability, we report per-iteration wall-clock times (mean ± sd over 15 runs) for the Alpine1 benchmark across varying dimensionalities, keeping the same Gumbel sampling and training parameters, with slightly more optimization trials for the acquisition maximization. Results are shown in Table S9, indicating feasible runtimes up to $d = 12$.

Sensitivity to perceptual-noise scale σ. To assess how misspecifying the perceptual-noise scale affects CPBO, we re-ran all seven benchmarks while fixing σ to $0.25\times$ and $4\times$ its default value. Table S10 shows that inference regret after 30 iterations is essentially unchanged across these settings. The middle column (σ) corresponds to the default setting reported in Table 1 in the main paper; the other two columns replicate that table under misspecified noise.

Table S9: Wall-clock time per BO iteration (seconds) as a function of dimensionality, averaged over 15 runs on Alpine1.

Dimension	d=2	d=3	d=6	d=10	d=12
Wall-clock (s)	193±19	229±22	345±34	510±43	626±50

To quantify this robustness, we performed paired two-sided Wilcoxon signed-rank tests comparing the default setting with each misspecified setting across runs. In all cases, we found no statistically significant differences ($p > 0.05$), indicating that performance is not sensitive to the precise choice of σ. When σ is under- or overestimated, other trainable quantities (notably the learned JND threshold $\hat{\gamma}$ and the latent-function scale) adapt to absorb the mismatch, leaving both acquisition behavior and final performance effectively stable.

Table S10: Sensitivity of CPBO$_{\hat{\gamma}}$ to the perceptual-noise scale σ: inference regret (mean ± sd) after 30 iterations with σ fixed to 0.25×, 1× (original, as in Table 1), and 4×.

$f(y)$	0.25σ	σ	4σ
Branin	.022 ± .021	.020 ± .017	.019 ± .017
Six-hump	.008 ± .006	.009 ± .004	.012 ± .007
Bohachevsky	.001 ± .002	.001 ± .001	.001 ± .001
Levy13	.012 ± .014	.011 ± .016	.009 ± .012
Bukin6	.131 ± .048	.115 ± .053	.119 ± .052
Cross-Tray	.143 ± .069	.150 ± .064	.146 ± .067
Ackley	.113 ± .048	.093 ± .052	.115 ± .048

Sensitivity to oracle-feedback noise. We also evaluate robustness to noise in the preference oracle itself. Unlike the sensitivity analysis for the model noise scale σ, this experiment changes the RUM noise used to generate pairwise feedback while keeping the experimental setup otherwise identical to Table 1. Figure S3 shows that CPBO$_{\hat{\gamma}}$ is consistently less affected by increasing oracle-feedback noise than the baselines.

We summarize this trend by computing the area under the inference-regret curve across noise levels. On Branin, CPBO has 42.1% lower AUC than EUBO and 36.6% lower AUC than POP-BO. On Bohachevsky and Six-hump, CPBO has 69.6% and 21.3% lower AUC than EUBO, respectively. These results suggest that modeling indifference through the learned threshold $\hat{\gamma}$ improves robustness when pairwise preferences become increasingly noisy.

Comparison with qEUBO. In our consecutive-comparison setup, each iteration evaluates one new candidate against a fixed reference, which effectively corresponds to a batch of size $q = 2$ in the original batch formulation. In this regime, qEUBO (Astudillo et al., 2023) reduces to a Monte Carlo estimator of the same expected-utility-of-best-outcome (EUBO) objective, for which closed-form expressions are available. Table S11 provides a direct comparison between qEUBO ($q = 2$), the analytic EUBO implementation, and CPBO$_{\hat{\gamma}}$, mirroring the Table 1 setup. As shown, both EUBO formulations yield nearly identical results, confirming their equivalence under this regime, while CPBO consistently achieves lower inference regret across all benchmark functions.

For completeness, we further evaluate qEUBO under a Gumbel-noise oracle calibrated to the "moderate logistic" level, as described in Astudillo et al. (2023). This setup mirrors the conditions of Table 1 but employs a different noise model. As summarized in Table S12, the overall conclusions remain unchanged: CPBO$_{\hat{\gamma}}$ achieves lower inference regret and higher ordinal accuracy than the baseline, under this alternative noise model.

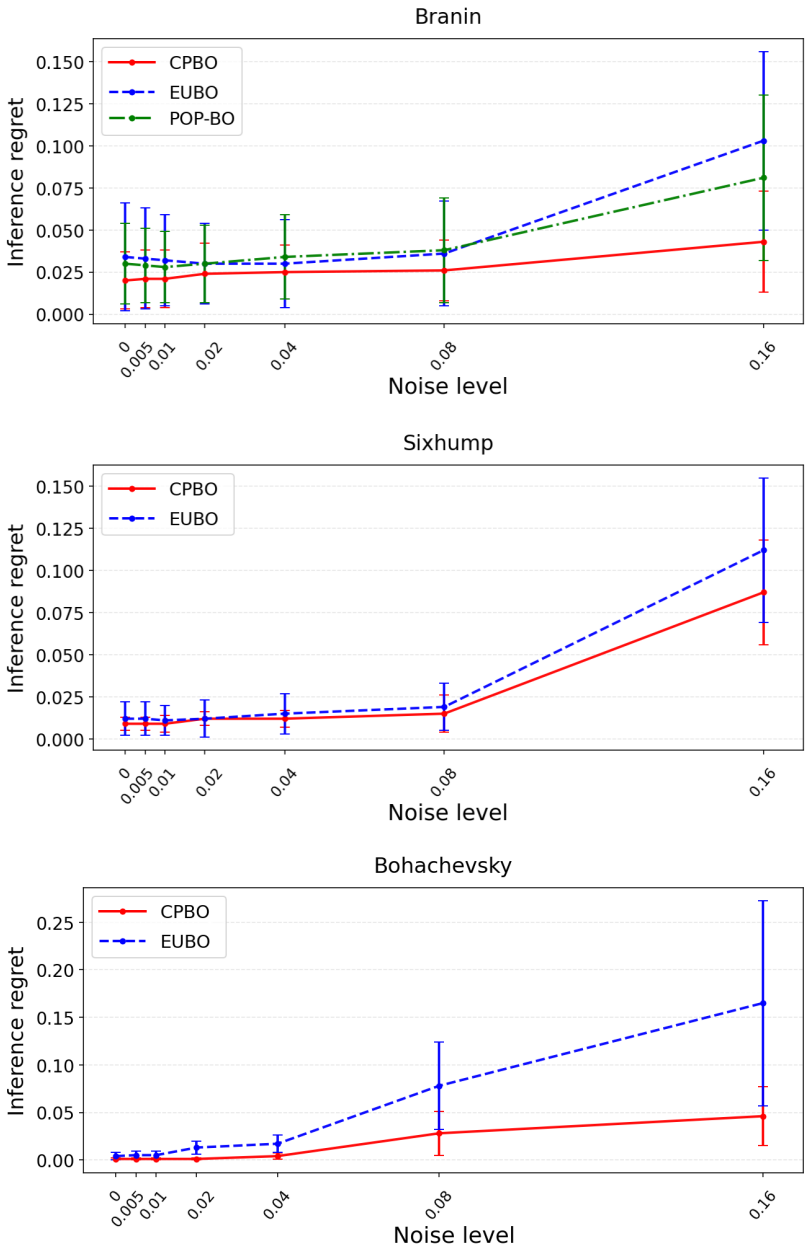

Figure S3: Sensitivity to oracle-feedback noise (RUM noise) on the three representative benchmark functions. We vary the RUM noise used to generate pairwise preferences while keeping the rest of the Table 1 setup fixed. The plots report inference regret after 30 iterations for CPBO, EUBO, and, where available, POP-BO. Markers show the mean across runs and vertical bars show ± standard deviation.

Table S11: Inference regret after 30 iterations (mean ± sd) for qEUBO ($q = 2$), EUBO, and CPBO$_{\hat{\gamma}}$, mirroring the format of Table 1.

$f(y)$	**qEUBO**	**EUBO**	**CPBO$_{\hat{\gamma}}$**
Branin	.036 ± .031	.034 ± .032	**.020 ± .017**
Six-hump	.014 ± .011	.012 ± .010	**.009 ± .004**
Bohachevsky	.004 ± .004	.004 ± .004	**.001 ± .001**
Levy13	.018 ± .013	.016 ± .012	**.011 ± .016**
Bukin6	.129 ± .081	.126 ± .078	**.115 ± .053**
Cross-Tray	.171 ± .108	.177 ± .112	**.150 ± .064**
Ackley	.151 ± .122	.149 ± .118	**.093 ± .052**

Table S12: Optimization and exploration performance measured by inference regret (Regret) and global ordinal accuracy of the utility proxy (Ord.) under a Gumbel-noise oracle calibrated to the "moderate logistic" level, mirroring the Table 1 setup but with qEUBO. Values are mean ± sd, with bold indicating the best method.

$f(y)$	**qEUBO**		**CPBO$_{\hat{\gamma}}$**	
	Regret↓	Ord.↑	Regret↓	Ord.↑
Branin	0.044 ± 0.035	64.4 ± 6.9	**0.023 ± 0.021**	**85.4 ± 3.4**
Six-hump	0.023 ± 0.019	68.9 ± 6.6	**0.010 ± 0.006**	**80.8 ± 3.6**
Bohachevsky	0.009 ± 0.007	75.6 ± 7.6	**0.001 ± 0.001**	**93.3 ± 2.0**
Levy13	0.025 ± 0.018	72.5 ± 6.8	**0.015 ± 0.016**	**83.0 ± 3.9**
Bukin6	0.141 ± 0.096	73.5 ± 6.1	**0.124 ± 0.061**	**92.8 ± 2.1**
Cross-Tray	0.175 ± 0.115	58.4 ± 6.8	**0.154 ± 0.069**	**64.9 ± 7.4**
Ackley	0.167 ± 0.128	69.2 ± 8.2	**0.101 ± 0.057**	**83.1 ± 3.8**

C.5 Additional Visualizations

This section provides visual illustrations to help interpret model behavior beyond the quantitative results. We focus on three aspects: (i) the estimated probability of indifference conditioned on a reference point, (ii) how indifference regions vary across utility functions and perceptual thresholds, and (iii) differences in the utility surfaces recovered by CPBO and competing baselines.

Recovered Indifference Probability. Figure S4 illustrates the probability of indifference under a fixed reference point (marked in red). The left image visualizes the ground-truth probability that a new candidate will be judged indifferent to the reference, based on the true utility function with perceptual noise and $\gamma_{\text{true}} = 0.1$. The right image shows the same quantity estimated from a trained CPBO model after 30 iterations. The similarity of the patterns indicates that CPBO successfully captures the shape and extent of the indifference region with limited observations.

Indifference Region. Figure S5 illustrates how indifference regions vary across utility functions, perceptual thresholds, and reference points. Since all utility functions are normalized to the $[0, 1]$ range, the global maximum corresponds to a reference utility of 1.0, while the mid-level reference is set to 0.5. These two points illustrate how the JND threshold γ_{true} translates into perceptual indifference depending on both the reference location and the underlying utility landscape. We compare a fixed threshold $\gamma_{\text{true}} = 0.1$ with a γ_{true} yielding $\mathbb{E}[\sim] = 30\%$. For ease of interpretation, we indicate the γ band directly, rather than the probability of an indifferent response ($r_t = 0$) as shown in Figure S4; the yellow areas here correspond to regions within that contour.

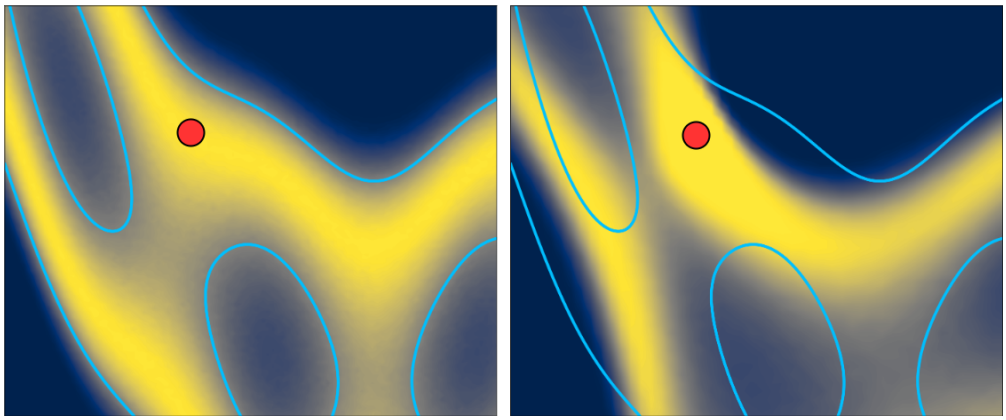

Figure S4: Indifference probability heatmaps conditioned on a candidate (red circle). (Left) Ground-truth probability based on $\gamma_{\text{true}} = 0.1$ and known utility. (Right) Estimated probability from CPBO after 30 iterations. The contour indicates the boundary of JND region.

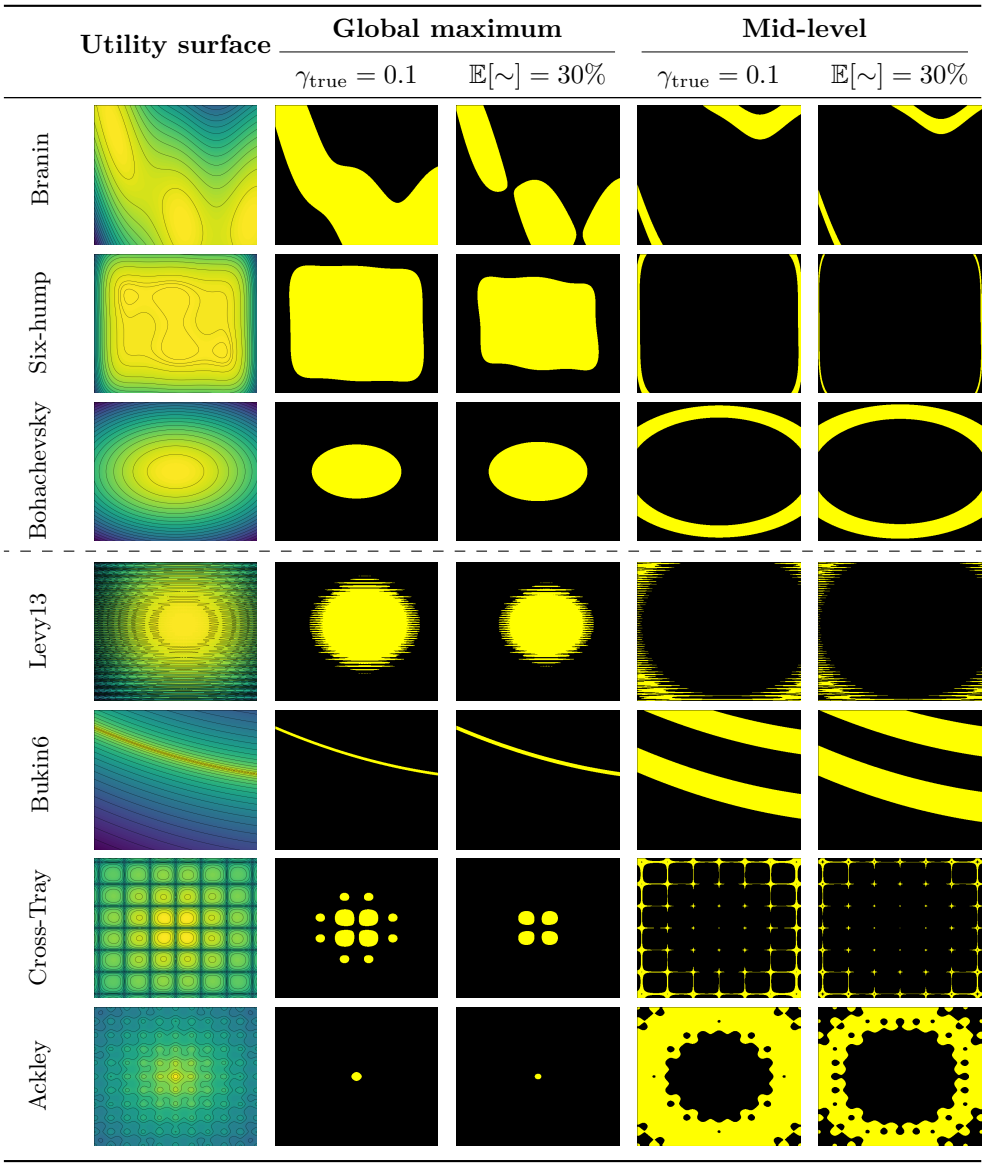

Figure S5: Indifference regions for seven benchmark utility functions. Each column shows areas within γ_{true} of a fixed reference point, using either a fixed threshold ($\gamma_{\text{true}} = 0.1$) or a threshold chosen to yield $\mathbb{E}[\sim] = 30\%$. Reference points correspond to the global maximum (1.0) and the mid-level utility (0.5) on the normalized $[0, 1]$ surface. The dashed line separates smoother, preferentially relevant functions (top) from non-smooth, spiky functions (bottom).

Utility Surface. Figure S6 shows example latent utility surfaces after 30 iterations on three benchmarks. While results may vary between runs, CPBO$_{\hat{\gamma}}$ generally provides more accurate reconstructions of the true utility compared to the baselines. The adapted MES acquisition promotes broader exploration, and incorporating a learnable JND threshold improves reconstruction in regions with flat or ambiguous preferences.

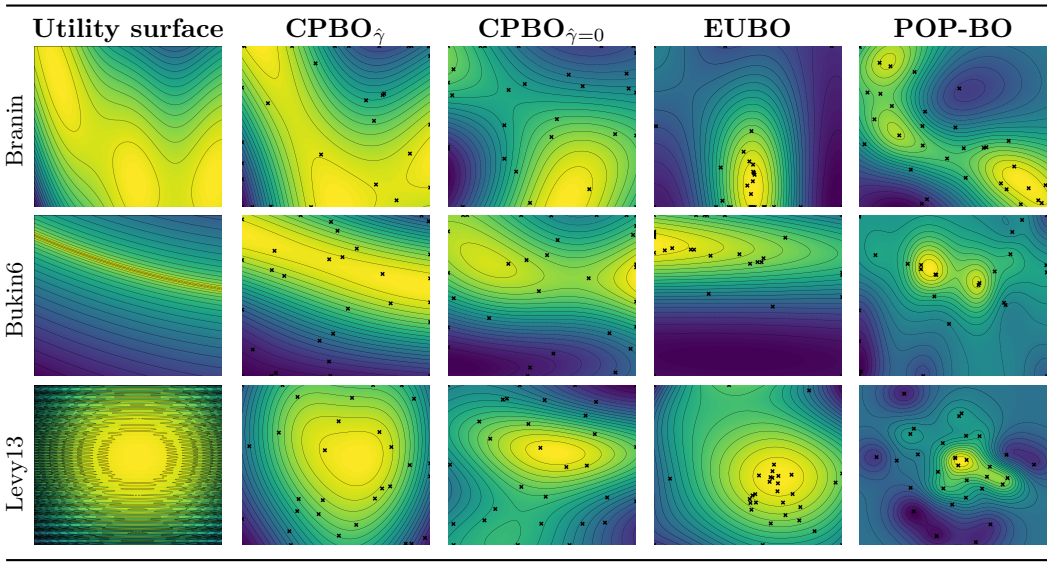

Figure S6: Latent surface examples for the three benchmark utilities under $\gamma_{\text{true}} = 0.1$ and 30 iteration training, comparing CPBO with estimated $\hat{\gamma}$, CPBO at $\hat{\gamma} = 0$, EUBO, and POP-BO.

D EXTRUSION OPTIMIZATION

This section describes the real-world optimization study used to evaluate CPBO in a practical setting. The task involves optimizing a physical process using human preferences as feedback. We provide an overview of the experimental setup, including operator instructions, the optimization task, and supporting tools for real-time interaction.

D.1 Overview

In this experiment, three qualified operators with varying levels of experience were tasked with solving a high-moisture extrusion optimization challenge using recommendations generated by the CPBO algorithm. A technical facilitator (one of the authors) assisted the process by recording verbal preference responses r_t from each operator, inputting them into the algorithm, and communicating the recommended configuration. Recommendations were typically displayed on a screen, and operators recorded them on their individual tracking sheets. A fourth operator was responsible for the physical preparation of the 15 initial configurations (see Section D.4 for details), which were then evaluated by the decision-making operators before proceeding to CPBO runs.

D.2 Operator Instructions

All participating operators were employed at the institute where the experiment was conducted and were already familiar with extrusion processing. They varied in their experience with optimization tasks and use of the specific extruder, ranging from novice/intermediate to advanced and expert levels. The advanced and expert operators operated the extruder independently. In contrast, the novice operator, who lacked hands-on experience with this particular extruder model, received technical support from an independent operator responsible for the initial configurations, but made all operational decisions independently.

For this study, operators received a shared set of task-specific instructions to ensure consistency. These included general guidance on how to interact with the CPBO system, evaluate samples, and provide preference feedback. The individual steps and instructions are paraphrased below, in a condensed format:

- The operator should compare each newly produced sample to the one produced in the previous step.
- After each production, the operator packages the produced sample, putting it into a zip-lock bag, for future evaluations.
- The preferences are to be based on the overall quality of the extrudate, considering characteristics such as fibril length, toughness, and tearing force. These characteristics are commonly used in the field and the operators are familiar with their meaning. A high-quality extrudate is defined as one that exhibits a fibrous structure resembling meat-like texture, but beyond this the operators were judged to provide their subjective evaluation on the overall quality.
- If the operator cannot distinguish a difference or considers both samples equally good, they report them as "the same", corresponding to the indifference option in CPBO.
- If the recommended configuration is expected to cause an extruder blockage or produce clearly flawed material, the operator stops the run and marks the new candidate as not preferred.
- Preferences are communicated verbally to the facilitator, who records and enters them into the optimization system.
- After each acquisition run, the newly recommended configuration is displayed on a screen for the operator to input into the extruder.

D.3 Optimization Task

The experiment was conducted using a twin-screw extruder (Process 11 Hygienic, screw diameter 11 mm, L/D 40:1) equipped with a long cooling die designed specifically for high-moisture extrusion of fibrous meat-analogue products. The goal was to identify optimal values for three parameters: temperature, screw speed, and water feed, jointly denoted as $x \in \mathbb{R}^3$, that yield a high-quality extrudate $y(x)$.

The task represents a subcomponent of a general R&D workflow, where the aim is to identify a sufficiently good configuration to proceed to downstream analysis and product evaluation. For each feed mixture candidate, we first need to find the optimal configuration before it makes sense to continue further.

In this study, the feed mixture consisted of 50% pea protein isolate and 50% hemp protein concentrate, and as explained above a high-quality extrudate is defined as one that exhibits a fibrous structure resembling meat-like texture.

D.4 Experiment Details

The study comprised one initial session and three operator-specific optimization sessions. In the initial phase, an independent operator produced $N_0 = 15$ initial extrudates by executing candidate configurations, selected via Latin hypercube sampling (McKay et al., 1979) over the 3D parameter space of temperature, screw speed, and water feed. Each configuration x_i was used to produce a physical sample $y(x_i)$.

All initial samples were frozen immediately after production. During the operator sessions, each operator received three 5 cm segments of each sample, which were reheated in sealed plastic cups at 40 °C for 10 minutes and then left at room temperature for 20 minutes before the evaluation.

Each operator evaluated the initial samples through pairwise comparisons, generating preference data $\mathcal{D}_t = \{([x_{i-1}, x_i], r_i)\}_{i=2}^{t}$. These responses were used to fit a surrogate model $f_0(x)$, serving as the starting point for the CPBO algorithm.

During the experiment, the surrogate model $f_t(x)$ was iteratively updated from the operator's feedback. At each step t, the operator produced the candidate $y(x_t)$ using the configuration x_t, and compared it to the previous candidate. The process continued for 30 total steps (15 initial + 15 interactive).

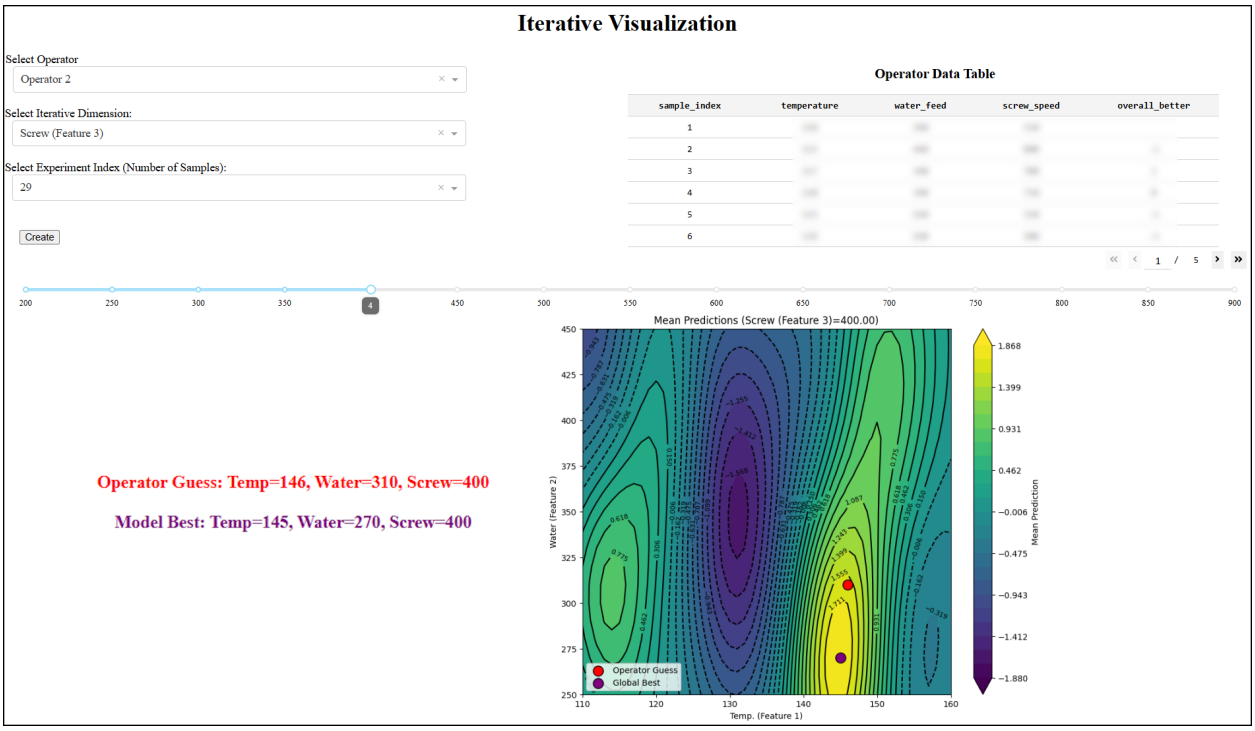

Figure S7: Web-based GUI developed with Plotly Dash for monitoring the optimization process. Tabular data is blurred for privacy.

At steps $t = 24$ and $t = 29$, the current surrogate model's maximizer $\hat{x}^* = \arg\max_x f_t(x)$ was selected for production, resulting in the 25th and 30th candidates, respectively. Unlike the other candidates, which were selected based on the acquisition function to maximize information gain, these were explicitly recommended by the model as the best configurations. Those configurations were retained for potential reproduction in sensory panel testing and further qualitative assessment.

Figure S7 shows a web-based interactive visualization tool developed using Plotly Dash. The interface allows tracking the model's predicted optimum over time and visualizing how the estimated utility evolves as more feedback is collected, providing insights into operator preferences.

D.5 Technical Details

The real-world extrusion optimization followed the same GP surrogate and acquisition configurations used in the synthetic benchmark experiments, with one exception: we found it beneficial to impose a prior over the GP kernel lengthscales. Specifically, we placed independent Gamma priors $\ell_d \sim \text{Gamma}(1.0, 0.05)$ on each lengthscale dimension.

Fixed Parameters. A constant flour feed of 300 g/h was maintained throughout the experiment. The temperature of the cooling die was set to 45 °C. The first four adjustable barrel zones were fixed at 60, 70, 80, and 100 °C, respectively.

Optimized Parameters. Three parameters were optimized during the experiment:

- **Barrel temperature** in the last four extruder zones: 110–160 °C, with 1 °C resolution
- **Water feed:** 250-450 ml/h, with 10 ml/h resolution

- **Screw speed:**: 200-900 rpm, with 50 rpm resolution

These ranges were selected based on prior empirical knowledge of stable process windows and known safe operating boundaries for the twin-screw extruder and the feed mixture. The temperature limits were set to enable the production of reasonable extrudates while avoiding a high risk of extruder blockage. The water feed limits ensured that the resulting moisture content was appropriate, given the fixed flour feed of 300 g/h. The chosen screw speed range corresponded to the practical operating range of the extruder.

D.6 Evaluation

After the experiment, the operators were informally interviewed to assess their satisfaction with the final recommendation produced by CPBO. They were also given the opportunity to continue the experiment by testing additional configurations of their own choosing, allowing them to explore whether they could identify a better solution through manual iteration. In addition, the operators were asked to comment on the fixed parameters and the selected boundaries for the optimized variables, to confirm their practical relevance for the task. Although we do not report individual responses, the interviews served to verify that the recommendations were satisfactory and that the experimental setup aligned with the operators' expectations.

D.7 Results

The three human operators demonstrated notably different preferences, converging to three distinct configurations. Figure S8 illustrates 2D slices of the utility surface, always fixing one setting to its optimal value and visualizing the estimated utility over the remaining two. Operator 1 preferred a balanced setting across temperature and water feed rate. Operator 2 found two distinct configurations acceptable, indicating flexibility in the desired product. In contrast, operator 3 strongly favored high water content with a specific temperature range. Despite reaching different solutions, all operators were satisfied with the final extrudate, highlighting the subjective nature of the task. Furthermore, the three operators indicated that the fixed parameters and the selected parameter boundaries were appropriate and consistent with their domain knowledge and expectations for stable operation.

The plot also shows the learned threshold γ indicating the configurations that the model considers likely to produce candidates indistinguishable from the optimal one for this operator. This information helps in validating sensitivity of the optimal solution.

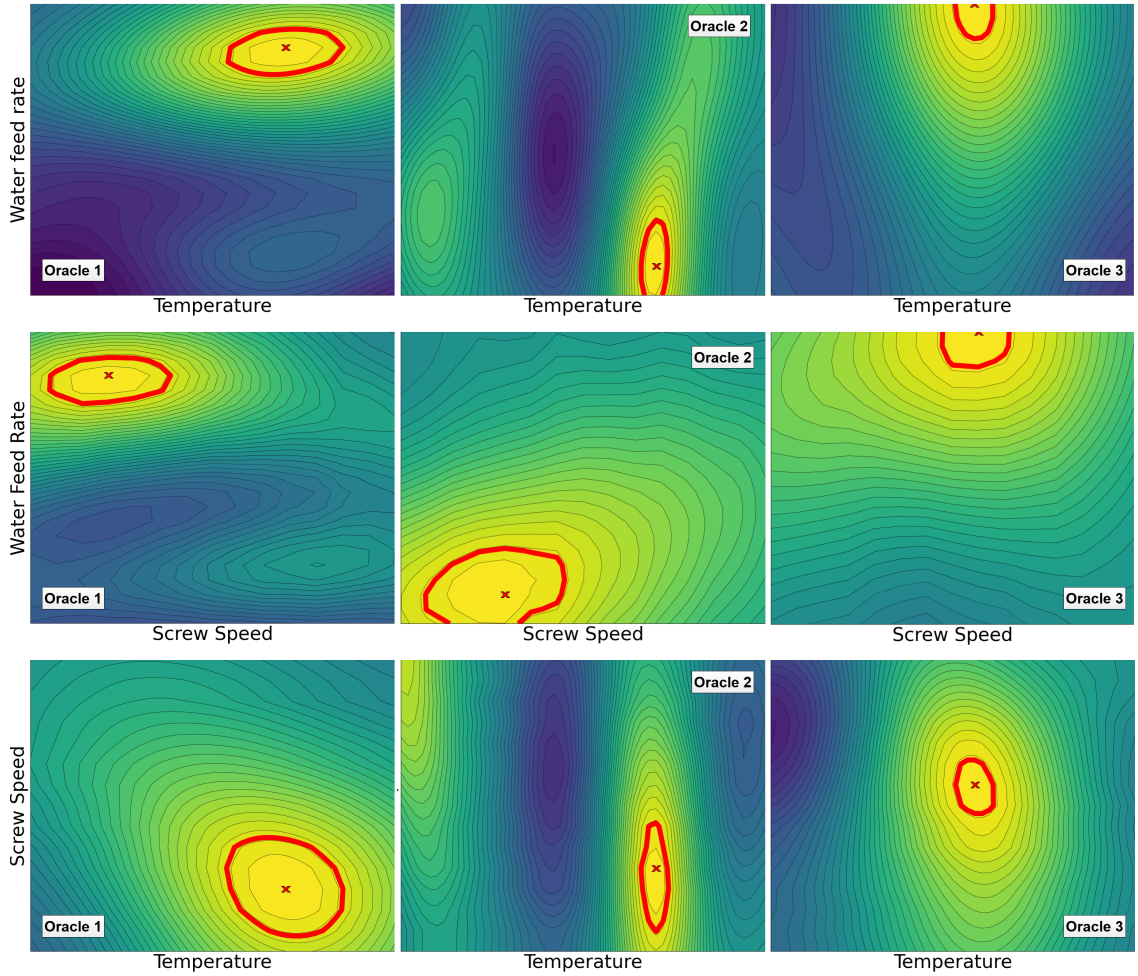

Figure S8: 2D slices of the learned utility surfaces for the three human operators, with the third parameter fixed to its estimated optimal value. Red 'x' marks the estimated optimum, and the bold red contour shows the learned JND region conditioned on the estimated optimum.