Coactive Preference-Guided Multi-Objective Bayesian Optimization: An Application to Policy Learning in Personalized Plasma Medicine

Shao, Ketong; Chakrabarty, Ankush; Mesbah, Ali; Romeres, Diego

TR2025-021 March 01, 2025

Abstract

The design of advanced learning- and optimization-based controllers requires selecting parameters that balance performance objectives and constraints. Bayesian optimization (BO) has proven effective for resource-efficient calibration of such controllers. Preference-guided BO incorporates user preferences to prioritize areas of interest, but it lacks a mechanism for users to specify desired outcomes directly. This paper introduces a user-centric framework for preference-guided BO, leveraging a novel knowledge- gradient based coactive acquisition function that allows users not only to select preferred outcomes but also also propose alternatives to guide exploration. To enable efficient implementation, we approximate the acquisition function, avoiding costly bilevel optimization. The approach is validated for control policy adaptation in personalized plasma medicine, where it outperforms standard preference-guided BO by effectively integrating user feedback to personalize treatment protocol.

IEEE Control Systems Letters 2025

Coactive Preference-Guided Multi-Objective Bayesian Optimization: An Application to Policy Learning in Personalized Plasma Medicine

Ketong Shao[1], Ankush Chakrabarty[2],*Senior Member, IEEE*, Ali Mesbah[1]*Senior Member, IEEE*, Diego Romeres[2], *Senior Member, IEEE*

***Abstract*—The design of advanced learning- and optimization-based controllers requires selecting parameters that balance performance objectives and constraints. Bayesian optimization (BO) has proven effective for resource-efficient calibration of such controllers. Preference-guided BO incorporates user preferences to prioritize areas of interest, but it lacks a mechanism for users to specify desired outcomes directly. This paper introduces a user-centric framework for preference-guided BO, leveraging a novel knowledge-gradient based coactive acquisition function that allows users not only to select preferred outcomes but also also propose alternatives to guide exploration. To enable efficient implementation, we approximate the acquisition function, avoiding costly bilevel optimization. The approach is validated for control policy adaptation in personalized plasma medicine, where it outperforms standard preference-guided BO by effectively integrating user feedback to personalize treatment protocol.**

***Index Terms*—Biomedical; Optimization; Human-in-the-loop control**

I. Introduction

CONTROLLER calibration can be challenging in real-world settings, particularly when controller parameters influence closed-loop performance and constraint satisfaction in highly non-convex ways. To address this challenge, Bayesian optimization (BO) [1] has emerged as an effective controller calibration technique, especially for learning-based and optimization-based control strategies wherein the closed-loop control performance is an implicit, black-box function of control policy parameters [2], [3].

Multi-objective BO (MOBO) has been developed to handle multiple, potentially conflicting, closed-loop performance objectives and constraint functions. MOBO expands the known Pareto front (PF) by selecting query points based on their potential to extend the PF or their expected information gain [4], [5]. However, the efficiency of MOBO can be limited, as it may expend effort exploring parts of the parameter space unlikely to yield desired outcomes.

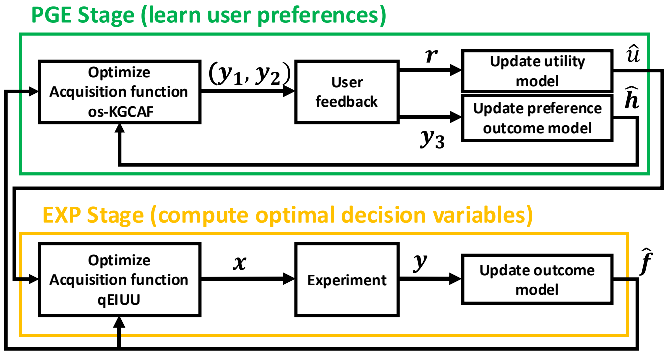
PGE Stage (learn user preferences)
Optimize Acquisition function os-KGCAF
(y_1, y_2)
User feedback
r
Update utility model
$\hat{u}$
Update preference outcome model
$\hat{h}$
y_3
EXP Stage (compute optimal decision variables)
Optimize Acquisition function qEIUU
x
Experiment
y
Update outcome model
$\hat{f}$

Fig. 1: Schematic of the two-staged User-Centric Coactive Preference-Guided Bayesian Optimization (uc-PGBO).

To better integrate user knowledge and preferences, preferential BO has been introduced [6]–[8]. These approaches learn a latent utility function in relation to design parameters, aiming to maximize the utility value of queries. However, they generally do not consider performance objectives directly, which can hinder their effectiveness. BO with preference exploration (BOPE) [9] addresses this limitation by employing two Gaussian process surrogates to learn the relationship between design parameters and performance objectives, and the mapping from objectives to a latent utility. Despite these advancements, existing approaches often constrain user input to pairwise comparisons between objective sets. While some methods allow users to rank multiple outcomes [10] or suggest hypothetical outcomes [11], they do not systematically incorporate this feedback into the acquisition function.

This paper presents a BO approach, termed user-centric coactive preference-guided BO (uc-PGBO), featuring a novel knowledge-gradient coactive acquisition function (KGCAF). Our proposed uc-PGBO enables users to not only select preferred outcomes from provided options but also suggest alternative outcomes. The KGCAF incorporates the knowledge of the proposed model for predicting user-proposed alternatives based on historical data of user suggestions and presented outcomes. This ensures that subsequent outcomes presented to the user are chosen while anticipating user-provided alternatives. To address the computational complexity inherent in knowledge-gradient acquisition functions due to

[1] KS and AM are with the Department of Chemical and Biomolecular Engineering, University of California, Berkeley, CA, USA. {ketong_shao, mesbah}@berkeley.edu

[2] AC and DR are with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. {chakrabarty, romeres}@merl.com

nested optimization, we introduce an approximation called one-shot KGCAF (os-KGCAF). This approximation eliminates the nested optimization structure through the use of two auxiliary variables, thereby enhancing the computational efficiency of uc-PGBO. uc-PGBO's main contributions are the KGCAF and os-KGCAF acquisition functions, whose effectiveness is demonstrated through a control policy learning task in personalized plasma medicine [12].

II. User-Centric CoactivePreference-Guided BO

We aim to solve a multi-objective optimization problem, which can be recast into a utility maximization problem:

$$\max_{\boldsymbol{x}\in\mathbb{X}} u\left(\boldsymbol{f}(\boldsymbol{x})\right), \tag{1}$$

where $\boldsymbol{x} \in \mathbb{X} \subset \mathbb{R}^d$ are decision variables, and $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^o$ represents the multiple-objectives, termed *outcomes*. This outcome function is expensive to evaluate and behaves as a black-box oracle, meaning gradients of $\boldsymbol{f}$ are unavailable. We assume certain outcomes are preferred by the user to others, and this is modeled by an unknown utility function $u : \mathbb{R}^o \to \mathbb{R}$. Noisy observations of the outcomes are available: $\boldsymbol{y}_i = \boldsymbol{f}(\boldsymbol{x}_i)+\epsilon$, where ϵ is independent Gaussian noise with variance $\sigma_\epsilon^2 I_o$. To solve (1), we employ BO techniques, referring to the approach as preference-guided BO (PGBO).

Algorithm 1: User-Centric Coactive Preference-Guided Bayesian Optimization (uc-PGBO)

```
Input: $\mathcal{D}_{N_0}, \mathcal{P}^c_{\tilde{N}_0}, \mathcal{Q}_{\tilde{N}_0}$
Output: $\mathcal{D}$
Parameter : $N_x, M, Q$
0. Initialization
Train $\hat{\boldsymbol{f}}(\cdot|\mathcal{D}_{N_0})$ the multi-outcome model;
Train $\hat{u}(\cdot|\mathcal{P}^c_{\tilde{N}_0})$ the utility model;
Train $\hat{\boldsymbol{h}}(\cdot|\mathcal{Q}_{\tilde{N}_0})$ the preference outcome model;
Set $\mathcal{D} = \mathcal{D}_{N_0}, \mathcal{P}^c = \mathcal{P}^c_{\tilde{N}_0}, \mathcal{Q} = \mathcal{Q}_{\tilde{N}_0}$;
for $q$=1:$Q$ do
    1. PGE Stage
    for $m$=1:$M$ do
        predict $\hat{\boldsymbol{y}}_3 = \hat{\boldsymbol{h}}(\hat{\boldsymbol{f}}(\boldsymbol{x}_1), \hat{\boldsymbol{f}}(\boldsymbol{x}_2))$;
        $\boldsymbol{x}_{1,m}, \boldsymbol{x}_{2,m} = \underset{\boldsymbol{x}_1,\boldsymbol{x}_2}{\text{argmax}}$ KGCAF$(\boldsymbol{x}_1, \boldsymbol{x}_2, \hat{\boldsymbol{y}}_3)$;
        $\boldsymbol{y}_{1,m}, \boldsymbol{y}_{2,m} = \hat{\boldsymbol{f}}(\boldsymbol{x}_{1,m}), \hat{\boldsymbol{f}}(\boldsymbol{x}_{2,m})$;
        $r(\boldsymbol{y}_{1,m}, \boldsymbol{y}_{2,m}) \leftarrow$ pairwise preference;
        $\boldsymbol{y}_{3,m} \leftarrow$ coactive feedback based on $\boldsymbol{y}_{1,m}, \boldsymbol{y}_{2,m}$;
        Update $\mathcal{P}^c = \mathcal{P}^c \cup \{p_m, c_{1,m}, c_{2,m}\}$;
        Update $\mathcal{Q} =$
          $\mathcal{Q}\cup\{((\boldsymbol{y}_{1,m}, \boldsymbol{y}_{2,m}), \boldsymbol{y}_{3,m}), ((\boldsymbol{y}_{2,m}, \boldsymbol{y}_{1,m}), \boldsymbol{y}_{3,m})\}$
        Train $\hat{u}(\cdot|\mathcal{P}^c)$ the utility model;
        Train $\hat{\boldsymbol{h}}(\cdot|\mathcal{Q})$ the preference outcome model;
    end
    2. EXP Stage
    $\boldsymbol{x}_{1:N_x,q} = \text{argmax}\, q\text{EIUU}(\boldsymbol{x}_{1:N_x,q})$ optimal decision
      variables;
    $\boldsymbol{y}_{1:N_x,q} = \boldsymbol{f}(\boldsymbol{x}_{1:N_x,q})$ outcome function evaluations
    Update $\mathcal{D} = \mathcal{D} \cup \{(\boldsymbol{x}_{i,q}, \boldsymbol{y}_{i,q})\}_{i=1}^{N_x}$;
    Train $\hat{\boldsymbol{f}}(\cdot|\mathcal{D})$ the outcome surrogate model;
end
```

A. User-centric Coactive PGBO (uc-PGBO) Framework

Herein, we describe our two-stage iterative method that alternates the *preference-guided exploration* (PGE) and the *experimentation* (EXP) stage.

PGE Stage: The PGE stage gathers information about user preferences to improve the utility function approximation. It maximizes the proposed knowledge-gradient co-active acquisition function (KGCAF) M times. In the m-th iteration, a pair of outcomes $\{(\boldsymbol{y}_{1,m}, \boldsymbol{y}_{2,m})\}$ is presented to the user, who provides a two components preferential feedback: (i) pairwise preference $r_m := r(\boldsymbol{y}_{1,m}, \boldsymbol{y}_{2,m}) \in \{1, 2\}$, indicating their preferred outcome; and (ii) coactive feedback $\boldsymbol{y}_{3,m}$, a suggested vector outcome they prefer over the observed $(\boldsymbol{y}_{1,m}, \boldsymbol{y}_{2,m})$. This feedback embeds user knowledge, guiding the algorithm towards better outcomes in future iterations. We assume that $\boldsymbol{y}_3$ is preferred, i.e., $u(\boldsymbol{y}_3) \geq \max\{u(\boldsymbol{y}_1), u(\boldsymbol{y}_2)\}$. With this feedback, two surrogate models are updated: (i) the utility model $\hat{u}(\cdot)$, which learns from the preference pair $p_m := \{(\boldsymbol{y}_{1,m}, \boldsymbol{y}_{2,m}), r_m\}$ and coactive feedback pairs $c_{1,m} := \{(\boldsymbol{y}_{3,m}, \boldsymbol{y}_{1,m}), 1\}$, $c_{2,m} := \{(\boldsymbol{y}_{3,m}, \boldsymbol{y}_{2,m}), 1\}$; (ii) the preference outcome model $\boldsymbol{h}(\cdot, \cdot)$, mapping $(\boldsymbol{y}_{1,m}, \boldsymbol{y}_{2,m})$ to $\boldsymbol{y}_{3,m}$. Coactive feedback has been studied before, c.f. [13], [14], however, in these works, the feedback $\boldsymbol{y}_3$ has been used to improve the utility model, without explicitly accounting for a preference outcome model $\boldsymbol{h}$. A novelty of this work is that $\boldsymbol{y}_3$ is explicitly used to update $\boldsymbol{h}$, which in turn embeds preference information into the maximization of the KGCAF; therefore the maximizer of the KGCAF will yield outcome pairs that should be strongly preferred by the user.

EXP Stage: This stage seeks decision variables $\boldsymbol{x}$ that yield high utility values, ideally converging to the optimal. The outcome function $\boldsymbol{f}$ is approximated by a surrogate model $\hat{\boldsymbol{f}}$, used to maximize the batch noisy expected improvement under utility uncertainty (qEIUU). This results in N_x decision variables $\{\boldsymbol{x}\}_{1:N_x}$, which are passed to $\boldsymbol{f}$ to obtain the actual outcome vectors $\boldsymbol{y}_{1:N_x}$. The outcome model is then updated with training pairs $(\boldsymbol{x}_{1:N_x}, \boldsymbol{y}_{1:N_x})$. The EXP stage is implemented as proposed in [9]. The PGE and EXP stages repeat for a finite budget of Q iterations, after which the user selects their preferred outcome and decision variables.

B. Modeling Outcome, Utility and Preference outcome

We describe the modeling of the outcome function $\boldsymbol{f}$, the utility function u and the preference outcome model $\boldsymbol{h}$ using Gaussian processes ($\mathcal{GP}s$) [15], [16] and pairwise $\mathcal{GP}s$ [17].

Outcome model: Each objective in $\boldsymbol{f}$ is modeled by an independent $\mathcal{GP}$, $f^a(\cdot) \sim \mathcal{N}(0, k^a(\cdot, \boldsymbol{x}'))$, $\forall a \in \{1, \ldots, o\}$ where k^a is the prior covariance (kernel). During the q-th iteration of the EXP stage, the dataset $\mathcal{D}^a$ is updated with $\{(\boldsymbol{x}_i, y_i^a)\}_{i=1}^N$, where $y^a = f^a(\boldsymbol{x}) + \epsilon$. The posterior at a new decision variable $\boldsymbol{x}^*$ becomes $\hat{f}^a \sim \mathcal{N}(\mu^a(\boldsymbol{x}^*), \Sigma^a(\boldsymbol{x}^*))$:

$$\mu^a(\boldsymbol{x}^*) = k_N^a(\boldsymbol{x}^*)(K_N^a + (\sigma_\epsilon^a)^2 I_N)^{-1}\boldsymbol{y}^a,$$

$$\Sigma^a(\boldsymbol{x}^*) = k^a(\boldsymbol{x}^*, \boldsymbol{x}^*) - k_N^a(\boldsymbol{x}^*)(K_N^a + (\sigma_\epsilon^a)^2 I_N)^{-1} k_N^a(\boldsymbol{x}^*)^\top,$$

where $k_N^a(\boldsymbol{x}^*) = [k^a(\boldsymbol{x}^*, \boldsymbol{x}_1), \ldots, k^a(\boldsymbol{x}^*, \boldsymbol{x}_N)]$ and $K_N^a \in \mathbb{R}^{N\times N}$ is the kernel matrix.

Utility model: The latent utility function u is modeled using pairwise $\mathcal{GP}$ based on relative information. Observations from user feedback $\mathcal{P}_n^c = \{p_j, c_{1,j}, c_{2,j}\}_{j=1}^n$ where n is the number of human feedback, are used at each PGE iteration. The pairwise comparisons follow a probit likelihood:

$$\mathbb{P}\left(r\left(\boldsymbol{y}_1, \boldsymbol{y}_2\right)=1 \mid u\left(\boldsymbol{y}_1\right), u\left(\boldsymbol{y}_2\right)\right)=\Phi\left(\frac{u\left(\boldsymbol{y}_1\right)-u\left(\boldsymbol{y}_2\right)}{\sqrt{2} \lambda}\right),$$

where λ is a hyperparameter and Φ is the normal CDF, as shown in [17]. The posterior of u is:

$$\mathbb{P}(u \mid \mathcal{P}_n^c) \propto \mathbb{P}(u) \prod_{j=1}^{n} \Phi\left(\frac{u(\boldsymbol{y}_{j,p})-u(\boldsymbol{y}_{j,np})}{\sqrt{2}\lambda}\right)$$
$$\Phi\left(\frac{u(\boldsymbol{y}_{j,3})-u(\boldsymbol{y}_{j,p})}{\sqrt{2}\lambda}\right)\Phi\left(\frac{u(\boldsymbol{y}_{j,3})-u(\boldsymbol{y}_{j,np})}{\sqrt{2}\lambda}\right),$$

where $\mathbb{P}(u)$ is the prior of the utility, $\boldsymbol{y}_{j,p}$ is the preferred outcome over the non preferred $\boldsymbol{y}_{j,np}$ outcome.

Preference outcome model: The user's outcome preference function $\boldsymbol{h} : \mathbb{R}^{2o} \rightarrow \mathbb{R}^{o}$, mapping $(\boldsymbol{y}_1, \boldsymbol{y}_2)$ to the preferred outcome $\hat{\boldsymbol{y}}_3$, is also modeled by $\mathcal{GP}$. It is trained after n coactive feedback using the dataset $\mathcal{Q}_n = \{((\boldsymbol{y}_{1,j}, \boldsymbol{y}_{2,j}), \boldsymbol{y}_{3,j}), ((\boldsymbol{y}_{2,j}, \boldsymbol{y}_{1,j}), \boldsymbol{y}_{3,j})\}_{j=1}^{n}$.

C. Algorithm

The uc-PGBO is outlined in Alg. 1 and visually represented in the flowchart in Fig. 1. At the start of the procedure decision variables $X_{N_0} = \{(\boldsymbol{x}_i)\}_{i=1}^{N_0}$ are sampled, either from a prior distribution when available or uniformly in $\mathbb{X}$. Given X_{N_0} the true outcome function $\boldsymbol{f}$ is evaluated to generate noisy observation that create the dataset $\mathcal{D}_{N_0} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N_0}$. From this dataset we generate $\tilde{N}_0$ random pairs and ask the user for the pairwise preference $\mathcal{P}^c_{\tilde{N}_0} = \{p_j, c_{1,j}, c_{2,j}\}_{j=1}^{\tilde{N}_0}$ and the coactive feedback $\mathcal{Q}_{\tilde{N}_0} = \{((\boldsymbol{y}_{1,j}, \boldsymbol{y}_{2,j}), \boldsymbol{y}_{3,j})\}_{j=1}^{\tilde{N}_0}$. Then, the outcome model $\hat{\boldsymbol{f}}$, the utility model $\hat{u}$ and the preference model $\hat{\boldsymbol{h}}$ are trained on each respective dataset. The number of BO loops, Q, the number of iterations within each PGE stage, M, and the number of optimal decision variables, N_x, generated in the EXP stage are defined as parameters. Subsequently, the PGE and EXP stage described in Sec. II-A are alternated for Q BO loops and the algorithm outputs a dataset $\mathcal{D}$ with the optimal decision variables and associated optimal outcomes that the user can choose from. Notice that, in practice, preference or coactive feedback may be omitted. However, this may deteriorate the performance, as KGCAF relies on accurate utility and preference outcome model.

III. Knowledge-Gradient Coactive AF

The classical knowledge gradient (KG) AF seeks to maximize information gain from adding a new sample (in this case, an outcome vector pair) by considering the difference in utility models before and after incorporating the new sample. To mitigate the computational complexity of evaluating KG, the expected upper bound optimization (EUBO) method approximates the KG [9]. However, EUBO considers users' pairwise feedback and, thus, is not suitable for handling users' coactive feedback, as described in Sec. II-A. To this end, we propose the knowledge gradient coactive acquisition function (KGCAF), which incorporates coactive feedback as

$$V_{\text{KGCAF}}(\boldsymbol{y}_1, \boldsymbol{y}_2)$$
$$= \mathbb{E}\Big[\max_{\boldsymbol{y}\in\mathcal{Y}} \mathbb{E}\big[u(\boldsymbol{y})\big|\mathcal{P}^c_{n+1}\big] - \max_{\boldsymbol{y}\in\mathcal{Y}} \mathbb{E}\big[u(\boldsymbol{y})\big|\mathcal{P}^c_{n}\big]\Big|\mathcal{P}^c_{n}\Big] \tag{2a}$$
$$\equiv \mathbb{E}\Big[\max_{\boldsymbol{y}\in\mathcal{Y}} \mathbb{E}\big[u(\boldsymbol{y})\big|\mathcal{P}^c_{n+1}\big]\Big] \tag{2b}$$
$$= \sum_{i=1}^{2} \mathbb{P}\big(r(\boldsymbol{y}_1, \boldsymbol{y}_2) = i\big|\mathcal{P}^c_{n,\boldsymbol{y}_3}\big) \max_{\boldsymbol{y}\in\mathcal{Y}} \mathbb{E}\big[u(\boldsymbol{y})\big|\mathcal{P}^c_{n+1}\big], \tag{2c}$$

where $\mathcal{P}^c_{n,\boldsymbol{y}_3} = \mathcal{P}^c_n \cup \big\{(\boldsymbol{y}_3, \boldsymbol{y}_1, 1), (\boldsymbol{y}_3, \boldsymbol{y}_2, 1)\big\}$ and

$$\mathcal{P}^c_{n+1} = \mathcal{P}^c_n \cup \big\{(\boldsymbol{y}_1, \boldsymbol{y}_2, r), (\boldsymbol{y}_3, \boldsymbol{y}_1, 1), (\boldsymbol{y}_3, \boldsymbol{y}_2, 1)\big\}. \tag{3}$$

Note that (2b) is equivalent from a optimization perspective since the term $\max_{\boldsymbol{y}\in\mathcal{Y}} \mathbb{E}\big[u(\boldsymbol{y})\big|\mathcal{P}^c_n\big]$ is independent of the optimization variables $\boldsymbol{y}_1, \boldsymbol{y}_2$, while (2c) is true because the expectation accounts for two cases: y_1 being preferred over y_2, and y_2 being preferred over y_1.

KGCAF aims to maximize the difference in expected maximum utility before and after the preference between outcomes $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ is revealed, considering an alternative outcome $\boldsymbol{y}_3$. However, KGCAF suffers from two challenges: it requires future information, i.e., $\boldsymbol{y}_3$, and involves a nested optimization that can be computationally prohibitive. We mitigate the first challenge by integrating a data-driven preference outcome model $\boldsymbol{h}$, which predicts $\hat{\boldsymbol{y}}_3$ based on $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. Then, to simplify the nested optimization, we propose a 'one-shot' approximation, termed os-KGCAF, which reduces computational complexity by solving a single maximization problem, rather than the nested optimization of KGCAF. The os-KGCAF is defined as

$$V^*_{\text{os-KGCAF}} = \max_{\boldsymbol{y}_1, \boldsymbol{y}_2, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2 \in \mathcal{Y}} V_{\text{os-KGCAF}}(\boldsymbol{y}_1, \boldsymbol{y}_2, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2), \tag{4}$$

where

$$V_{\text{os-KGCAF}} = \mathbb{P}(r=1|\mathcal{P}^c_{n,\hat{\boldsymbol{y}}_3})\mathbb{E}[u(\tilde{\boldsymbol{y}}_1)|\mathcal{P}^c_{n,\hat{\boldsymbol{y}}_3} \cup \{(\boldsymbol{y}_1, \boldsymbol{y}_2, 1)\}]$$
$$+ \mathbb{P}(r=2|\mathcal{P}^c_{n,\hat{\boldsymbol{y}}_3})\mathbb{E}\big[u(\tilde{\boldsymbol{y}}_2)|\mathcal{P}^c_{n,\hat{\boldsymbol{y}}_3} \cup \{(\boldsymbol{y}_1, \boldsymbol{y}_2, 2)\}\big] \tag{5}$$

In (5), two auxiliary decision variables $\tilde{\boldsymbol{y}}_1$ and $\tilde{\boldsymbol{y}}_2$ are optimized concurrently with $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, enabling the AF to be computed without solving expensive inner optimizations. The computation of the os-KGCAF is described in Alg. 2. In effect, os-KGCAF approximates KGCAF, significantly reducing computation by focusing on only two possible preferences between $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, thus requiring just two auxiliary variables.

Let $\boldsymbol{y}^\star \triangleq \arg\max_{\mathcal{Y}} u(y)$ denote the optimally-preferred function value. We look to show $\boldsymbol{y}^\star$ is attainable by maximizing the os-KGCAF (5) as $n \to \infty$. To show this, we must first define a finite-sample approximation of (2) as follows

$$\hat{V}_{\text{KGCAF}} = \frac{1}{N_n} \sum_{i=1}^{N_n} \max_{\boldsymbol{\zeta}_i \in \mathcal{Y}} \mathbb{E}[u(\boldsymbol{\zeta}_i) \mid \mathcal{P}^c_{n+1}] - \Xi(\boldsymbol{y}). \tag{6}$$

Here, $\boldsymbol{\zeta}_{1:N_n}$ are auxiliary variables, $\Xi(\boldsymbol{y}) = \max_{\boldsymbol{y}\in\mathcal{Y}} \mathbb{E}[u(\boldsymbol{y}) \mid \mathcal{P}^c_n]$ is independent of $\boldsymbol{\zeta}_{1:N_n}$. Note that $\mathcal{P}^c_{n+1}$ depends on $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, as defined in (3). We observe that (6) is a nested optimization problem that can be recast as a tractable optimization problem, c.f. [18], as

$$\boldsymbol{y}_{1:2}, \boldsymbol{\zeta}_{1:N_n} = \text{argmax} \frac{1}{N_n} \sum_{i=1}^{N_n} \mathbb{E}[u(\boldsymbol{\zeta}_i) \mid \mathcal{P}^c_{n+1}(\boldsymbol{y}_{1:2})], \tag{7}$$

where all $\zeta_{1:N_n}$ and $\boldsymbol{y}_{1:2}$ are solved concurrently; hence, referred to as a 'one-shot' approach. We show that at the limit $n \to \infty$ we can recover $u(\boldsymbol{y}^\star)$ almost surely, under some smoothness assumptions on the utility.

Proposition 1: Suppose $\mathcal{Y}$ is compact and $u(\cdot)$ has a $\mathcal{GP}$ prior with continuously-differentiable mean and covariance functions. Let $\mathcal{P}^c_n$ be the dataset obtained via maximization of a finite-sample approximation of os-KGCAF and the corresponding coactive feedback $\{\boldsymbol{y}_{3,j}\}_{j=1}^{n}$. Then $u(\boldsymbol{y}_n) \rightarrow u(\boldsymbol{y}^\star)$ as $n \rightarrow \infty$ (a.s.), with $\boldsymbol{y}_n \triangleq \arg\max_{\boldsymbol{y}\in\mathcal{Y}} \hat{u}\big(\boldsymbol{y}|\mathcal{P}_n^c\big)$.

Proof: Consider a finite-sample approximation of (5) defined as

$$\hat{V}_{\text{os-KGCAF}} = \frac{1}{N_n}\sum_{i=1}^{N_n} \mathbb{E}[u(\boldsymbol{y}_i)|\mathcal{P}^c_{n,\hat{\boldsymbol{y}}_3} \cup \{(\boldsymbol{y}_1, \boldsymbol{y}_2, r_i)\}], \quad (8)$$

where N_n is the number of approximating samples. Given that $r \in \{1, 2\}$ it follows from (8) that

$$\tilde{V} = \frac{N_{n|r=1}}{N_n}\mathbb{E}\left[u(\tilde{\boldsymbol{y}}_1)|\mathcal{P}^c_{n,\hat{\boldsymbol{y}}_3} \cup \{(\boldsymbol{y}_1, \boldsymbol{y}_2, r = 1)\}\right] + \frac{N_{n,|r=2}}{N_n}\mathbb{E}\left[u(\tilde{\boldsymbol{y}}_2)|\mathcal{P}^c_{n,\hat{\boldsymbol{y}}_3} \cup \{(\boldsymbol{y}_1, \boldsymbol{y}_2, r = 2)\}\right], \quad (9)$$

where $N_{n|r=1}$ and $N_{n|r=2}$ represent the count of instances among the N_n samples where $\hat{u}_1 \geq \hat{u}_2$ and vice versa. As $N_n \rightarrow \infty$, we recover

$$\tfrac{N_{n|r=1}}{N_n} \rightarrow \mathbb{P}\left(r = 1 \mid \mathcal{P}^c_{n,\hat{\boldsymbol{y}}_3}\right), \tfrac{N_{n|r=2}}{N_n} \rightarrow \mathbb{P}\left(r = 2 \mid \mathcal{P}^c_{n,\hat{\boldsymbol{y}}_3}\right),$$

corresponding to the probabilities in (2c) and (5). Thus, (8) asymptotically tends to (6) as $N_n \rightarrow \infty$. Let $\bar{\mathcal{P}}_n^c$ be the dataset obtained via maximization of os-KGCAF in (8) and the corresponding coactive feedback $\{\boldsymbol{y}_{3,j}\}_{j=1}^{n}$ and define $\bar{\boldsymbol{y}}_n \triangleq \arg\max_{\boldsymbol{y}\in\mathcal{Y}} \hat{u}\big(\boldsymbol{y}|\bar{\mathcal{P}}_n^c\big)$. We now invoke [18, Theorem 4], by which if N_n is chosen such that $\limsup_{n\in\mathbb{N}} N_n = \infty$, we can compute $\bar{\boldsymbol{y}}_n$ that yields $u(\bar{\boldsymbol{y}}_n) \rightarrow u(\boldsymbol{y}^\star)$ a.s. Finally, since $\bar{\boldsymbol{y}}_n \rightarrow \boldsymbol{y}_n$ as $N_n \rightarrow \infty$ then $u(\boldsymbol{y}_n) \rightarrow u(\boldsymbol{y}^\star)$ a.s. ■

Remark 1: By introducing two auxiliary variables $\tilde{\boldsymbol{y}}_{1,2}$, the time complexity of KGCAF, originally $\mathcal{O}(k_1 \cdot ((k_2 + k_3 + 1) \cdot n^2 + 3n^3))$, is reduced in os-KGCAF to $\mathcal{O}(k \cdot (3n^3 + 3n^2))$. Here, $k_1 \in \mathbb{N}^+$ is the number of iterations to optimize (2) on $\boldsymbol{y}_1, \boldsymbol{y}_2$, and $k_2, k_3 \in \mathbb{N}^+$ are the iterations for the two independent maximization in (2c), while k is the total iteration count for os-KGCAF in (4). In the experiments we verified that k is typically below the sum of k_1, k_2, and k_3, which means $k < (k_1 + k_2 + k_3) \ll k_1(k_2 + k_3 + 1)$. Therefore, os-KGCAF, avoiding the multiplicative effect of the nested loops in KGCAF, leads to a significant reduction of the computational complexity. This aligns with measured wall-clock times showing os-KGCAF an order of magnitude faster than KGCAF while maintaining comparable performance.

IV. Case Study: Personalized Plasma Medicine

We demonstrate the performance of Algorithms 1 and 2 on an example plasma medicine application [12], where the goal is to personalize the plasma treatment protocol based on user feedback over a sequence of treatments. Specifically, we aim to adapt the parameters of a model predictive controller (MPC) that is used to control the thermal effects of a kHz-excited atmospheric pressure plasma jet in helium on a target substrate [19]. The dynamics of the plasma jet are given by

$$s_{k+1} = As_k + Ba_k + w_k, \quad (10)$$

where k is discrete-time step; the state consists of maximum surface temperature T (°C) and total optical intensity of plasma I (a.u.), $s = [T, I]^\top$; the input consists of helium flow rate q (SLM) and applied power P (W) to generate plasma, $a = [P, q]^\top$; and the state-space matrices A and B are identified via subspace identification, as detailed in [12]. The control objective is to control the cumulative thermal

Algorithm 2: One-shot Knowledge Gradient Coactive Acquisition Function

```
Input: $\hat{\boldsymbol{f}}, \hat{u}, \hat{h}, \boldsymbol{x}_1, \boldsymbol{x}_2, \tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2, \mathcal{P}_n^c$
Output: $V_{\text{os-KGCAF}}$
$\boldsymbol{y}_1, \boldsymbol{y}_2, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2, = \hat{\boldsymbol{f}}(\boldsymbol{x}_1), \hat{\boldsymbol{f}}(\boldsymbol{x}_2), \hat{\boldsymbol{f}}(\tilde{\boldsymbol{x}}_1), \hat{\boldsymbol{f}}(\tilde{\boldsymbol{x}}_2)$;
$\hat{\boldsymbol{y}}_3 = \hat{\boldsymbol{h}}(\boldsymbol{y}_1, \boldsymbol{y}_2)$;
Update $\mathcal{P}^c_{n,\hat{\boldsymbol{y}}_3} = \mathcal{P}_n^c \cup \{(\hat{\boldsymbol{y}}_3, \boldsymbol{y}_1, r = 1), (\hat{\boldsymbol{y}}_3, \boldsymbol{y}_2, r = 1)\}$
Train $\hat{u}_{\hat{\boldsymbol{y}}_3}(\cdot|\mathcal{P}^c_{n,\hat{\boldsymbol{y}}_3})$ the utility model ;
*Compute the joint distribution of the utility values of $\hat{u}^1$ and $\hat{u}^2$ under $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$: $p(\hat{u}^1, \hat{u}^2) = \mathcal{N}(\hat{\boldsymbol{\mu}}, \hat{\Sigma})$*
    For $\mathcal{GP}$ properties given $\hat{u}_{\hat{\boldsymbol{y}}_3}(\cdot|\mathcal{P}^c_{\hat{\boldsymbol{y}}_3})$, $\boldsymbol{y}_1$, $\boldsymbol{y}_2$ compute
    $\hat{\boldsymbol{\mu}} = [\mu_{\hat{u}^1}, \mu_{\hat{u}^2}]^\top$, $\hat{\Sigma} = [\hat{\sigma}_{11}, \hat{\sigma}_{12}; \hat{\sigma}_{21}, \hat{\sigma}_{22}]$
Build the normal distribution of
  $\hat{u}^1 - \hat{u}^2 \sim \mathcal{N}(\mu_{\hat{u}^1} - \mu_{\hat{u}^2}, \hat{\sigma}_{11} + \hat{\sigma}_{22} - \hat{\sigma}_{12} - \hat{\sigma}_{21})$;
Calculate $\mathbb{P}(r = 2|\mathcal{P}^c_{n,\hat{\boldsymbol{y}}_3}) = \Phi(\frac{\mu_{\hat{u}^2} - \mu_{\hat{u}^1}}{\hat{\sigma}_{11} + \hat{\sigma}_{22} - \hat{\sigma}_{12} - \hat{\sigma}_{21}})$ and
  $\mathbb{P}(r = 1|\mathcal{P}^c_{n,\hat{\boldsymbol{y}}_3}) = 1 - \mathbb{P}(r = 2|\mathcal{P}^c_{n,\hat{\boldsymbol{y}}_3})$;
Train two $\mathcal{GP}$ models $\hat{u}_{\boldsymbol{y}_1 \succ \boldsymbol{y}_2}$ and $\hat{u}_{\boldsymbol{y}_2 \succ \boldsymbol{y}_1}$, using
  $\mathcal{P}^c_{\hat{\boldsymbol{y}}_3} \cup \{\boldsymbol{y}_1, \boldsymbol{y}_2, r = 1\}$ and $\mathcal{P}^c_{\hat{\boldsymbol{y}}_3} \cup \{\boldsymbol{y}_1, \boldsymbol{y}_2, r = 2\}$,
  respectively;
Obtain mean functions of two $\mathcal{GP}$s: $\mu_{\hat{u}_{\boldsymbol{y}_1 \succ \boldsymbol{y}_2}}$, $\mu_{\hat{u}_{\boldsymbol{y}_2 \succ \boldsymbol{y}_2}}$;
Return
$V = \mathbb{P}(r = 1|\mathcal{P}^c_{n,\hat{\boldsymbol{y}}_3})\mu_{\hat{u}_{\boldsymbol{y}_1 \succ \boldsymbol{y}_2}}(\tilde{y}_1) + \mathbb{P}(r =$
  $2|\mathcal{P}^c_{n,\hat{\boldsymbol{y}}_3})\mu_{\hat{u}_{\boldsymbol{y}_2 \succ \boldsymbol{y}_2}}(\tilde{y}_2)$;
```

effects of plasma on a substrate, as quantified by the so-called cumulative equivalent minutes (CEM) measure [19]

$$\text{CEM}_{k+1} = \text{CEM}_k + K^{(T_{\text{ref}} - T_k)}\delta t, \quad (11)$$

where $K > 0$ is an exponential base related to physical characteristics of the target, $T_{\text{ref}} = 43$°C is a reference temperature, and $\delta t = 0.5$ s is the sampling interval. Accordingly, the following MPC problem is formulated

$$\min_{s(k),a(k)} \left(\text{CEM}_{sp} - \text{CEM}\left(N_p|k\right)\right)^2 \quad (12a)$$

$$\text{s.t. } s(i + 1|k) = \hat{A}s(i|k) + \hat{B}a(i|k), \quad (12b)$$

$$\text{CEM}(i + 1|k) = \text{CEM}(i|k) + \hat{K}^{(T_{\text{ref}} - T(i|k))}, \quad (12c)$$

$$(s(i|k), a(i|k)) \in \mathcal{S} \times \mathcal{A}, \quad s(0|k) = s(k), \quad (12d)$$

$\forall i \in \{0, \dots, N_p - 1\}$, where $s(k) = [s(0|k), \dots, s(N_p|k)]^\top$ and $a(k) = [a(0|k), \dots, a(N_p - 1|k)]^\top$ are sequences of predicted states and control actions over a prediction horizon $N_p = 5$; CEM_{sp} is a desired CEM setpoint that dictates the plasma treatment efficacy; $\mathcal{S} = [25°\text{C}, 0 \text{ a.u.}] \times [35°\text{C}, 80 \text{ a.u.}]$ and $\mathcal{A} = [1.5 \text{ W}, 1.5 \text{ SLM}] \times [5 \text{ W}, 5 \text{ SLM}]$ define the state and control constraints; and (12b) and (12c) represent a model of the true dynamics (10) and (11). The resulting MPC policy $\pi(s_k, \boldsymbol{x}) = a_0^*$ is a function of the parameters $\boldsymbol{x} = [\hat{A}_{11}, \hat{A}_{12}, \hat{A}_{21}, \hat{A}_{22}, \hat{K}]^\top$, which comprise the decisions in Alg. 1. The MPC problem (12) is implemented in Python using CasADi [20] and solved with IPOPT [21].

To ensure safe and effective plasma treatments, three intertwined treatment performance objectives must be taken into account. The first is to minimize the treatment time τ_p to reach the predefined CEM_{ref}. The second is to ensure that the maximum surface temperature T does not surpass a prespecified threshold $T_{\max}$, which serves as a safety-critical constraint. Given that the spatial distribution of CEM can extend beyond the target area [22], the third aspect is to set a threshold temperature $T_{\text{ref,outer}}$ for areas that should remain unaffected by the plasma treatment. This threshold would mitigate patient discomfort, or potential harm to regions outside the target area. Hence, the plasma treatment performance objectives can be formulated as

$$f_1(\boldsymbol{x}) = \tau_p, \quad f_2(\boldsymbol{x}) = \sum_{k=0}^{N} (T(k) - T_{\max})^2, \tag{13a}$$

$$f_3(\boldsymbol{x}) = \max_k 2\pi \int_{r_0}^{r_1} r(T_{\text{ref,outer}} - T_r(k))dr, \tag{13b}$$

where $T_{\max} = 45$°C; $T_{\text{ref,outer}} = 43.5$°C; $r_0 = 0.5$ mm and $r_1 = 2$ mm delineate the boundary between treated and untreated areas; and $T_r(k)$ describes the spatial distribution of surface temperature at time step k: $T_r(k) = (T(k) - T_{\inf})\, e^{-\frac{r^2}{\sigma^2}} + T_{\inf}$, with $T_{\inf} = 37$ °C and σ capturing the spread of temperature distribution, which is affected by the maximum surface temperature T and helium flow rate q [22]. We seek to balance the objectives in (13) based on patient feedback. We prioritize a 30-second treatment completion time without violating the safety-critical constraint on $T_{\max}$, while mitigating patient discomfort in non-target areas. Thus, the utility function is defined as

$$u = -\alpha|f_1 - 30| - \beta|f_2| - \gamma|f_3|, \tag{14}$$

with weights $\alpha = 1$, $\beta = 1000$, and $\gamma = 100$, and it is used to provide the pairwise feedback in Alg. 1. Additionally, we model a probability of providing incorrect pairwise preference, r, to reflect potential user's judgment errors. Alg.1, Step 12, provides an incorrect feedback with probability 5% when $|u(y_1) - u(y_2)| = 0.05$ and this probability increases as the difference between utilities of the outcomes decreases.

Since quantifying f_3 directly based on a patient's experience may not be straightforward in practice, in this work, f_3 is characterized by the "physician" in terms of the measured maximum surface temperature T and applied helium flow rate q, which govern the spatial distribution of surface temperature [22] and thus f_3. As such, the observations of objectives utilized for training the outcome model, $\hat{\boldsymbol{f}}$, and the utility model, $\hat{u}$, as well as capturing patient's feedback, are consolidated into $\boldsymbol{y} = [f_1, f_2, T, q]^\top$. Accordingly, the coactive feedback from a patient proposing a better alternative $\boldsymbol{y}_3$ than both $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ is defined as

$$\boldsymbol{y}_3 = \boldsymbol{y}^+ + \zeta\Delta\boldsymbol{y}, \quad \boldsymbol{y}^+ = [\boldsymbol{y}_1, \boldsymbol{y}_2]_{\mathcal{I}_{\boldsymbol{y}}}, \tag{15a}$$

$$\Delta\boldsymbol{y} = [\boldsymbol{y}^* - \boldsymbol{y}_1, \boldsymbol{y}^* - \boldsymbol{y}_2]_{\mathcal{I}_{\boldsymbol{y}}}, \tag{15b}$$

where $\mathcal{I}_{\boldsymbol{y}} = \text{argmin}\{\text{abs}(\boldsymbol{y}^* - \boldsymbol{y}_1), \text{abs}(\boldsymbol{y}^* - \boldsymbol{y}_2)\}$ is the index vector from element-wise argmin, i.e., both argmin and abs functions applied element-wise. The improvement vector $\boldsymbol{y}^+$ is derived from the vectors as per $\mathcal{I}_{\boldsymbol{y}}$. The factor ζ affects the extent of proximity of the new suggestion $\boldsymbol{y}_3$ to $\boldsymbol{y}^*$, as well as to the initially proposed $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. With ζ set to 0.3 and $\boldsymbol{y}^* = [30, 0, 45, 1.5]^\top$, we desire a utility $u = 0$ in plasma treatments.

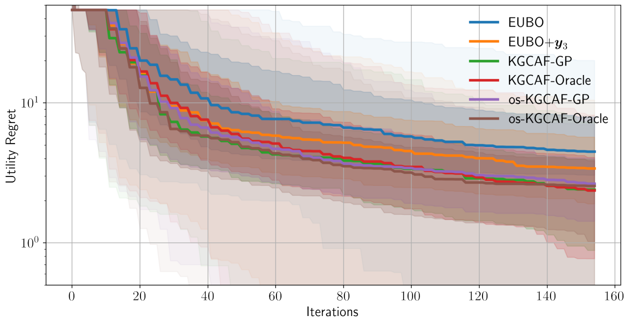

Fig. 2: Utility regret. Solid lines: Median across 256 runs, deep/light shades are quartiles/5-95 confidence intervals.

We consider the problem of adapting the parameters of the MPC (12) to tailor the treatment protocols to an individual patient. That is, the model parameters in (12b) are established for a population of patients, whereas we aim to adapt the parameters $\boldsymbol{x}$ in (12) over a series of treatments to personalize the treatment protocol to an individual patient while accounting for their feedback encoded by the utility (14); see [12] for the model specifications. To evaluate the performance of uc-PGBO, we consider a maximum treatment time of 120 s with the setpoint $\text{CEM}_{\text{sp}} = 1.5$ min. Note that the plasma treatment is stopped once CEM_{sp} is reached. The four entries in $\boldsymbol{y}$ are observed after each full treatment, and are modeled as independent $\mathcal{GP}$s. In Algorithm 1, the number of pairwise comparsions in the PE stage is $M = 1$, and the number of experiments in the EXP stage is $N_x = 3$. Generally, selecting large values for N_x or M carries the risk of making decisions in the absence of accurate surrogate models. The models $\hat{\boldsymbol{f}}$ and $\hat{u}$ are initially trained with $N_0 = 5$ samples $\mathcal{D}_{N_0} = \{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}_{i=1}^5$ obtained via uniform random sampling, and $\tilde{N}_0 = 3$ pairwise comparisons $\mathcal{P}^c_{\tilde{N}_0} = \{p_j, c_{1,j}, c_{2,j}\}_{j=1}^3$ and co-active comparsions $\mathcal{Q}_{\tilde{N}_0} = \{((\boldsymbol{y}_{1,j}, \boldsymbol{y}_{2,j}), \boldsymbol{y}_{3,j}), ((\boldsymbol{y}_{2,j}, \boldsymbol{y}_{1,j}), \boldsymbol{y}_{3,j})\}_{j=1}^3$.[1]

For comparison, we consider the following six preference-guided BO (PGBO) approaches: (i) PGBO with EUBO [9] (EUBO), (ii) PGBO with EUBO and $\boldsymbol{y}_3$ feedback (EUBO+$\boldsymbol{y}_3$), (iii) uc-PGBO with KGCAF, $\boldsymbol{y}_3$ feedback and $\hat{\boldsymbol{h}}$ modeled as $\mathcal{GP}$ (KGCAF-$\mathcal{GP}$), (iv) uc-PGBO with KGCAF, $\boldsymbol{y}_3$ feedback and the true (15) (KGCAF-Oracle), (v) uc-PGBO with one-shot KGCAF, $\boldsymbol{y}_3$ feedback and $\hat{\boldsymbol{h}}$ modeled as $\mathcal{GP}$ (os-KGCAF-$\mathcal{GP}$), and (vi) uc-PGBO with one-shot KGCAF, $\boldsymbol{y}_3$ feedback and the true (15) (os-KGCAF-Oracle). For uc-PGBO using KGCAF, the inner optimization iteration number is restricted to 100 to ensure its solution in a reasonable time; allowing unrestricted inner optimization, the solution computation time could take several hours. Each method is repeated 256 times, with different $\mathcal{D}_{N_0}$, $\mathcal{P}^c_{\tilde{N}_0}$ and $\mathcal{Q}_{\tilde{N}_0}$ used to initialize the outcome, utility and preference outcome surrogate model, and with $Q = 50$ BO loops of the PE and EXP stages.

Fig. 2 shows the utility regret in relation to the $155 := N_0 + Q \times N_x$ number of iterations for the six PGBO approaches. EUBO shows the highest regret due to ignoring $\boldsymbol{y}_3$. The diminished regret of EUBO+$\boldsymbol{y}_3$ indicates the benefit of introducing the coactive feedback (even while maintaining the same AF), which accelerates regret minimization by guiding the search process more effectively. uc-PGBO based on KGCAF and os-KGCAF markedly surpass the performance of EUBO+$\boldsymbol{y}_3$. However, the performance of the uc-PGBO based on KGCAF exhibits variability; surprisingly, KGCAF-$\mathcal{GP}$ outperforms KGCAF-Oracle that uses the true coactive feedback (15) in the initial iterations. This anomaly could stem from limited

[1] All implementations are done in BoTorch [18]. Training of multi-output $\mathcal{GP}$ $\hat{\boldsymbol{f}}$, pairwise $\mathcal{GP}$ $\hat{u}$ and preference $\mathcal{GP}$ $\hat{\boldsymbol{h}}$ is based on the default settings.

iterations in the inner optimization, where the $\mathcal{GP}$ surrogate of $\hat{\boldsymbol{h}}$ introduces a level of stochasticity to the predicted $\hat{\boldsymbol{y}}_3$. On the other hand, os-KGCAF-Oracle outperforms os-KGCAF-$\mathcal{GP}$, as expected. Overall, comparing the performance of PGBO approaches using a $\mathcal{GP}$ surrogate of $\hat{\boldsymbol{h}}$ versus the true Oracle information suggests that precise prediction of $\boldsymbol{y}_3$ may not be as critical, especially when the optimal user-defined $\boldsymbol{y}$ lies outside the Pareto frontier. Finally, the accuracy of os-KGCAF is demonstrated by an average normalized absolute difference of 0.023 between the optimal values of KGCAF and os-KGCAF, indicating a relatively minor discrepancy. In terms of computational efficiency, the average optimization time is 406.3 seconds for KGCAF, compared to the faster 58.0 seconds for os-KGCAF.

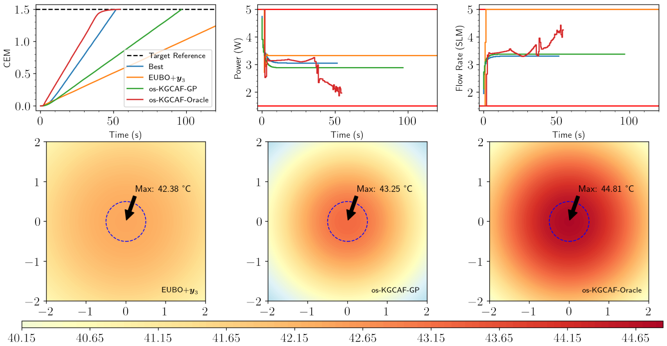

Fig. 3: Closed-loop control trajectories for MPC parameter adaptation using EUBO+$\boldsymbol{y}_3$ and uc-PGBO with os-KGCAF-$\mathcal{GP}$ and os-KGCAF-Oracle. Trajectories represent the highest utility from the least successful trial among 256 iterations. Top row: Time profiles of CEM, plasma power, and helium flow rate, with the blue line (os-KGCAF-Oracle) as the baseline. Bottom row: Surface temperature distributions for EUBO+$\boldsymbol{y}_3$, os-KGCAF-$\mathcal{GP}$, and os-KGCAF-Oracle.

Fig. 3 shows the closed-loop control trajectories for the least favorable achieved best final utility across 256 trials, comparing EUBO+$\boldsymbol{y}_3$, uc-PGBO with os-KGCAF-$\mathcal{GP}$, and os-KGCAF-Oracle. The blue line, representing the "best" profile from os-KGCAF-Oracle, serves as the baseline with the shortest treatment time while satisfying the comfort constraint (13). This worst-case scenario are highlighted because the most favorable achieved best final utilities are comparable across approaches. However, both os-KGCAF-$\mathcal{GP}$ and os-KGCAF-Oracle demonstrate superior performance over EUBO+$\boldsymbol{y}_3$, which fails to complete the treatment within 120 seconds. Further analysis of the five least successful trials indicates that both os-KGCAF-$\mathcal{GP}$ and os-KGCAF-Oracle consistently outperform EUBO+$\boldsymbol{y}_3$. uc-PGBO with os-KCGAF tends to achieve faster treatment times without exceeding temperature limits, while maintaining the surface temperature distribution within the target area. That is, the MPC policy parameters are adapted more effectively to maintain higher surface temperature and, thus, faster thermal dose delivery while honoring the safety- and comfort-related constraints on surface temperature.

V. Conclusions

This paper introduces a preference-guided BO framework with a novel knowledge-gradient-based acquisition function that integrates two types of user feedback: pairwise preference comparisons and coactive feedback. To ensure efficient implementation, the acquisition function is approximated to avoid the computational cost of bilevel optimization. Closed-loop simulations of MPC policy learning in personalized plasma medicine show that the proposed approach outperforms alternative preference-guided BO methods.

References

[1] J. Močkus, "On Bayesian methods for seeking the extremum," in *Optimization Techniques IFIP Technical Conference*. Springer, 1975, pp. 400–404.

[2] J. A. Paulson, F. Sorourifar, and A. Mesbah, "A tutorial on derivative-free policy learning methods for interpretable controller representations," in *Proceedings of the American Control Conference*. IEEE, 2023, pp. 1295–1306.

[3] A. Chakrabarty, "Optimizing closed-loop performance with data from similar systems: A bayesian meta-learning approach," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022.

[4] G. Makrygiorgos, A. D. Bonzanini, V. Miller, and A. Mesbah, "Performance-oriented model learning for control via multi-objective bayesian optimization," *Computers & Chemical Engineering*, 2022.

[5] M. Turchetta, A. Krause, and S. Trimpe, "Robust model-free reinforcement learning with multi-objective bayesian optimization," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 702–10 708.

[6] B. Eric, N. Freitas, and A. Ghosh, "Active preference learning with discrete choice data," *Advances in neural information processing systems*, vol. 20, 2007.

[7] J. González, Z. Dai, A. Damianou, and N. D. Lawrence, "Preferential bayesian optimization," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1282–1291.

[8] E. Siivola, A. K. Dhaka, M. R. Andersen, J. González, P. G. Moreno, and A. Vehtari, "Preferential batch bayesian optimization," in *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2021, pp. 1–6.

[9] Z. J. Lin, R. Astudillo, P. Frazier, and E. Bakshy, "Preference exploration for efficient bayesian optimization with multiple outcomes," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 4235–4258.

[10] M. Abdolshah, A. Shilton, S. Rana, S. Gupta, and S. Venkatesh, "Multi-objective bayesian optimisation with preferences over objectives," *Advances in neural information processing systems*, 2019.

[11] R. Ozaki, K. Ishikawa, Y. Kanzaki, S. Suzuki, S. Takeno, I. Takeuchi, and M. Karasuyama, "Multi-objective bayesian optimization with active preference learning," *arXiv preprint arXiv:2311.13460*, 2023.

[12] K. J. Chan, G. Makrygiorgos, and A. Mesbah, "Towards personalized plasma medicine via data-efficient adaptation of fast deep learning-based MPC policies," in *Proceedings of the American Control Conference*, 2023, pp. 2769–2775.

[13] M. Tucker, E. Novoseller, C. Kann, Y. Sui, Y. Yue, J. W. Burdick, and A. D. Ames, "Preference-based learning for exoskeleton gait optimization," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020.

[14] K. Shao, D. Romeres, A. Chakrabarty, and A. Mesbah, "Preference-guided bayesian optimization for control policy learning: Application to personalized plasma medicine," in *NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World*.

[15] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[16] A. Carè, R. Carli, A. Dalla Libera, D. Romeres, and G. Pillonetto, "Kernel methods and gaussian processes for system identification and control: A road map on regularized kernel-based learning for control," *IEEE Control Systems Magazine*, vol. 43, no. 5, pp. 69–110, 2023.

[17] W. Chu and Z. Ghahramani, "Preference learning with gaussian processes," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 137–144.

[18] M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy, "Botorch: A framework for efficient monte-carlo bayesian optimization," *Advances in neural information processing systems*, vol. 33, pp. 21 524–21 538, 2020.

[19] D. Gidon, D. B. Graves, and A. Mesbah, "Effective dose delivery in atmospheric pressure plasma jets for plasma medicine: A model predictive control approach," *Plasma Sources Science and Technology*, vol. 26, no. 8, p. 085005, 2017.

[20] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, 2019.

[21] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.

[22] D. Gidon, D. B. Graves, and A. Mesbah, "Predictive control of 2D spatial thermal dose delivery in atmospheric pressure plasma jets," *Plasma Sources Science and Technology*, vol. 28, no. 8, 2019.