Metacognition on Human-Machine Complementarity: Learning Multi-Dimensional Latent Human Preferences

Anonymous submission

Abstract

The human-machine complementarity refers to the complementary strengths and weaknesses of humans and machine algorithms in many real-world hybrid human-machine systems, such as semi-autonomous driving, industrial and manufacturing production lines. If machines' metacognitive assessments of the factors influencing complementarity and their own knowledge are accurate, allowing machines to opt in to human preference or expertise has the potential to achieve complementarity continuously. In this work, we propose a novel Bayesian learning framework for metacognition enhancement by capturing the latent temporal correlation of multi-dimensional human preferences. A crucial aspect of the framework is to guarantee robustness in changing environments across different system objectives, for which we formalize the sequential decision-making problem as a new constrained Gaussian best-arm identification problem, where preferences are modeled in implicit constraint functions. To solve the problem, we design the Dynamic Multi-Constraints Learning (DMCL) algorithm. DMCL relies on the Gaussian Process (GP) model where a new kernel function is designed to efficiently capture the latent temporal correlation. By employing a *Dynamic Constraint Selection strategy* (DCS), DMCL integrates the essential preferences that significantly contribute to complementarity into the machine's decision-making process to reduce human involvement. On the theoretical front, we prove the sublinear upper bounds of both cumulative regret and cumulative constraint violation. The regret bound further ensures the condition of accessibility to the optimal solution. On the experimental side, the results obtained from two experimental environments verify that our framework outperforms baseline methods in learning human preferences with minimum human involvement.

Introduction

With the wide deployment of Artificial Intelligence (AI) technologies, there is a growing trend in building hybrid systems that facilitate collaboration between humans and AI-embedded machines. The hybrid systems have led to superior performance in many practical applications, especially those with high risks, such as autonomous driving (Lindner et al. 2022) and medical diagnosis (Patel et al. 2019). Although previous research in this field has contributed to the methodologies of human-machine collaboration, such as human-in-the-loop machine learning (Liu et al. 2019; Ustalov, Fedorova, and Pavlichenko 2022; Elmalaki 2021), imitation learning from demonstration (Wu et al. 2019), and ensemble methods in machine learning (Patel et al. 2019)(Kerrigan, Smyth, and Steyvers 2021), there still exists an unclear understanding of human-machine complementarity.

To investigate the influence factors of human-machine complementarity, Steyvers *et al.* (Steyvers et al. 2022) develop a Bayesian framework to characterize the bound of accuracy differences between human and machine classifiers and then guide hybrid combinations of both. Focusing on human mental models, Kelly *et al.* (Kelly et al. 2023) propose to apply a cognitive science method to understand how humans perceive AI and optimize collaboartion performance in question-answering tasks. In this work, we examine the potential of human-machine complementarity for sequential decision-making problems, particularly in the input-driven environments where system dynamics are affected by exogenous stochastic input processes (Mao et al. 2018).

Our target is to improve machines' metacognitive assessments of how human preferences influence human-machine complementarity. However, it is intractable to characterize high-dimensional human preferences in the practical environments evolving in a time-varying and unknown way. For example, passengers expect autonomous vehicles to ensure safety and comfort while striving to minimize travel time. It is difficult to explicitly specify safety and comfort, which can only be assessed through passenger feedback. Lindner *et al.* in (Lindner et al. 2022) first propose to address that using the multi-arm bandit learning framework. The system objective is taken as a known linear reward function while the human preference is modeled as an unknown constraint function constructed from observations. However, the linear bandit settings cannot well-parameterize multi-dimensional and non-additive human preferences.

We consider more general bandit settings by modeling the system objective as an unknown general reward function and characterizing the stochastic characteristics of multi-dimensional preferences through a number of unknown general constraint functions. The formulation above categorizes our work into the realm of *Constrainted Stochastic Bandit Optimization* (CSBO) (Sui et al. 2015)(Berkenkamp, Krause, and Schoellig 2021)(Sui et al. 2018)(Amani, Alizadeh, and Thrampoulidis 2020)(Berkenkamp, Krause, and Schoellig 2021)(Zhou and

Ji 2022)(Xu et al. 2023)(Wachi and Sui 2020)(Berkenkamp, Krause, and Schoellig 2021)(Xu et al. 2023).

CSBO provides a promising formulation way to handle the black-box problems in which both reward and constraint are modeled as black-box functions. To find optimal solutions to various CSBO problems, the black-box functions are usually characterized by surrogate models built by samples from the Gaussian processes based on an assumption that the functions are bounded in reproducing kernel Hilbert space (RKHS). Among the works related to our settings, authors design various algorithms with theoretical bounds on cumulative regret or safety constraint by focusing on coupling the system objective and a single constraint (Sui et al. 2015), separating the objective and multiple constraints (Sui et al. 2018)(Berkenkamp, Krause, and Schoellig 2021), bandits with hard constraints (Amani, Alizadeh, and Thrampoulidis 2020), bandits with soft constraints (Zhou and Ji 2022), infeasibility declaration of optimization problem (Xu et al. 2023). Although these works provide important insights for learning human preferences as constraints, we need further enable machines to assess whether the constraints are necessary to be satisfied in order to achieve human-machine complementarity.

Motivation & Contribution

In summary, our work is motivated by two practical questions: *(1) How to examine the latent impacts of multi-dimensional human preferences on hybrid systems within input-driven environments; (2) How to improve machines' metacognitive assessments about the preferences for achieving human-machine complementarity during the interactive decision-making processes as many preferences are not always necessary.*
Our contributions are three-fold:

- **Problem Formulation** we formalize the human-machine task as a multi-constraints-based best-arm identification (MCBAI) problem, which improves the robustness and transferability of learned preferences since they are modeled as expensive-to-evaluate constraints. With the CGBAI problem, we are able to exploit both the temporal correlation of each preference and the combined impact of preferences and improve the decision-making capabilities of machines.
- **Algorithm Design** Technically, we design a new kernel function, extended from *Intrinsic Coregionalization Model* (ICM), which enables DMCL to efficiently examine the multi-dimensional and latent preferences with provably sublinear bounds of cumulative regret and cumulative constraint violation. With the new DSC, DMCL is able to learn the most essential preferences and integrate the knowledge into the decisions, accordingly achieving human-machine complementarity with minimum involvement.
- **Experiments** We evaluate DMCL's capability in learning human preferences through extensive experiments in two different environments, i.e., a two-link robot arm system and a autonomous driving system. We theoretically and empirically verify that DMCL can achieve efficient preference learning with minimum involvement compared to baseline methods.

Problem Statement

This section illustrates the formulation of the CGBAI problem in detail.

Definition 1. The CGBAI problem $\nu = (\mathcal{X}, r, c, \tau)$ consists of an unknown reward function $r \in \mathbb{R}$, a number of unknown constraint functions $\{c_i \in \mathbb{R}, \forall i \in [N]\}$. We define a finite integer set $[N] := 1, 2, \cdots, n \subset \mathbb{Z}$, corresponding to the dimension of human preferences. At each decision-making round, the machine determines a discrete action $x_t \in \mathcal{X}$ towards the maximization of $r(x_t)$ with respect to the satisfaction of constraints $c_i(x_t) \geq \tau_i$. Here, $\mathcal{X} \subset \mathcal{R}^d$ refers to a finite and discrete action set. Formally, the CGBAI problem is defined as:

$$\max_{x \in \mathcal{X}} r(x), \text{subject to } \; c_i(x) \geq \tau_i, \forall i \in [N]. \tag{1}$$

Throughout interactions with the environment, the machine's goal is to determine the optimal actions that satisfy $x_t^* = \arg\max_{x_t \in \mathcal{X}} r(x_t)$ at each decision-making round. In the literature related to multi-arm bandit problems, the action in (1) refers to the term *arm*.

Regularity Assumptions. To solve the CGBAI problem, it is essential to make standard assumptions concerning both r and c_i. Based on (Sui et al. 2015)(Brochu, Cora, and De Freitas 2010), the specific regularity assumptions are shown as follows.

The action set $\mathcal{X}$ is endowed with a positive definite kernel function. The CGBAI problem is assumed to be solvable, i.e., there exists at least one arm that satisfies the constraints with a safety tolerance. After assuming the reward function r has a bounded norm in the reproducing kernel Hilbert space (RKHS), it can be constructed by samples from the GP model $GP(\mu(x), k(x, x'))$, where the GP prior is characterized by a zero mean. Each observation is perturbed by noises with known noisy distribution, i.e., $n_{0,t} \sim N(0, \sigma^2)$. That is, the machine obtains the noisy observation $y_{0,t} = r(x_t) + n_{0,t}$ after identifying x_t. The same assumptions are also set for constraint functions $\{c_i, \forall i \in [N]\}$. The noisy observations obtained by the machine are denoted as $y_{i,t} = c_i(x_t) + n_{i,t}, \forall i \in [N]$.

By collecting T observations after identifying $X_T = [x_1, ...x_T]^T$, the GP posterior over the unknown functions can be built. Considering the temporal characteristics and combined impacts of multi-dimensional human preferences, we employ the multi-task GP (Bonilla, Chai, and Williams 2007) to build the surrogate models of the unknown functions in (1).

Regret and Violation. This part defines two metrics for measuring the performance of the CGBAI problem.

The first metric is *cumulative regret*, which refers to the cumulative gap between the step reward obtained by x_t and the oracle reward obtained by x_t^*. Since the oracle is not known, the goal of the CGBAI problem becomes how to minimize the reward gap progressively. Cumulative regret is formally defined as

$$R_T = \sum_{t=1}^{T} r(x_t^*) - r(x_t), \tag{2}$$

where $r(x_t^*)$ is the oracle reward while $r(x_t)$ refers to the step rewrad. T is a time horizon for the set of actions $X_T = [x_1, \cdots, x_T]^T$.

During the exploration phase, we permit violations of constraints, while in the subsequent optimization phase, we guarantee that the learned preferences adhere to the thresholds of constraints.

We define the second metric to measure the number of constraint violations in the exploration phase. The *constraint violation* indicating the violation of i-th constraint is denoted as $v_{i,t} = [\tau_i - c_i(x_t)]^+$, where $[\cdot]^+ := \max\{0, \cdot\}$. The *cumulative violation* is defined as:

$$V_{N,T} = \sum_{i=1}^{N} \sum_{t=1}^{T} v_{i,t}, \tag{3}$$

where N refers to the number of constraint functions.

The definitions of cumulative regret and cumulative violation guide our algorithm design. We next present the DMCL algorithm that achieves sublinear bounds of cumulative regret and cumulative violation.

Algorithm Design

In this part, we elaborate on the details of the DMCL algorithm. Before presenting the high-level description of DMCL, we introduce two innovations, i.e., the ICM-CI kernel and DCS.

ICM kernel with Contextual Information(ICM-CI)

In this part, the design principle of the new ICM-CI kernel is presented by extending the ICM kernel from (Bonilla, Chai, and Williams 2007). Recall that the CGBAI problem includes one reward function r and a set of independent constraint functions $\{c_i, \forall i \in [N]\}$, which forms a collection of unknown functions as in MTGP(Alvarez et al. 2012):

$$f_i(x) = \begin{cases} r(x) & i = 0 \\ c_i(x) & \forall i \in [N]. \end{cases} \tag{4}$$

Here, we use the index i to separate the reward function and constraint functions in f with an output space of $\mathbb{R}^{N+1}$. The ICM kernel, as a key component of MTGP, employs a covariance matrix to enable tasks to share a common underlying correlation structure, denoted as:

$$K((i,x),(i',x')) = B \otimes K(x,x'), \tag{5}$$

where B is a positive semi-definite (PSD) matrix that specifies the inter-task similarities, $K(x,x')$ is a covariance function shared across tasks.

In Bayesian Optimization, including additive contextual information is an efficient method to design useful algorithms in various real-world applications. For instance, in the multi-arm bandit settings (Krause and Ong 2011), authors integrate both continuous and discrete context spaces into the GP model and demonstrate superior performance compared to traditional context-free approaches. Motivated by this principle, we design the new ICM-CI kernel by incorporating contextual information related to multi-dimensional constraints into the model (5).

Specifically, we consider N dynamic contexts $\{z_i, i \in [N]\}$, where each context is associated with the information of the i-th constraint and evolves in response to environmental dynamics. Mathematically, we define each context as the ratio between the number of satisfied constraints and the total number of constraints, in order to indicate the results of constraint satisfaction and violation. The ratio is defined as:

$$z_i = \frac{\mathrm{count_i}}{T} \text{ for } i \in [N], \tag{6}$$

where T is the time steps used to indicate the number of iterations, $\mathrm{count_i}$ represents the iteration number at which the constraint i is satisfied.

By incorporating $\{z_i, i \in [N]\}$ into the model in (5), the new ICM-CI kernel is formulated as

$$K((i,x,z),(i',x',z')) = K((i,x),(i',x')) \otimes K(z,z'). \tag{7}$$

Hence, we incorporate the contextual information of constraints in the ICM-CI kernel, which enables the machine to acquire the patterns of changed reward and constraint functions with respect to various contexts.

Since the contextual information determined by the environmental dynamics is usually time-varying, the definition of the ICM-CI kernel with the covariance function improves the DMCL's capability in capturing the temporal correlation of human preferences.

Dynamic Constraints Selection Strategy (DCS)

Recall that our second goal of the DMCL algorithm is to capture the combined impact of multi-dimensional preference and then learn the most essential preferences with minimum human involvement. In this part, we introduce the design of DCS.

According to CGBAI (1), learning human preferences relies on sufficient observations obtained by a large volume of samples from GP models. In what follows, we consider how to minimize human involvement by reducing the number of samples. To accomplish that, we develop a criterion for evaluating the value of constraints. We first present three assumptions to specify the conditions of the ICM-CI kernel and DCS.

Assumption 1: The reward function and constraint functions are assumed to be mutually independent, which is implicitly reflected in the kernel definition (7).

Assumption 2: The same covariance function is shared among the reward function and constraints functions, implying that the smoothness of the response surface with respect to the parameters are consistent across finctions.

Assumption 3: We assume that the constraint weight distribution is dynamic and unknown, and our algorithm aims to infer it based on sequential observations.

The constraints' contributions to DMCL are specified in two aspects. First, constraints expand the region with safe actions, which thus narrows down the range of uncertain actions and speeds up the search process. Second, constraints contribute to the reward function according to the covariance function defined in the ICM-CI kernel. That is, constraints meeting the aforementioned two conditions have the potential to contribute to the preference learning of DMCL. Thus, the value of constraint i at time t can be examined by the following formula:

$$\text{weight}_{i,t}(x) = \alpha \frac{|S_{i,t}| - |S_{i,t} \bigcap S_{i,t-1}|}{|S_{i,t-1}|} + (1-\alpha)\delta_{0i} \quad (8)$$

where $S_{i,t}$ refers to the safe set with respect to the constraint i, defined as $\bigcup_{x \in S_{t-1}} \{x' \in X \mid l_{i,t}(x,z) \geq \tau_i\}$. $|S_{i,t} \bigcap S_{i,t-1}|$ represents the number of arms shared between the current feasible set and the feasible set obtained in the previous iteration. The term $\delta_{0i} = B_{0i}$ reflects the degree of correlation between c_i and r . Then, we adopt a normalization method to convert the constraint values into weights at each time t.

$$\text{weight}^*_{i,t}(x) = \frac{\text{weight}_{i,t}(x) - \text{weight}_{min,t}(x)}{\text{weight}_{max,t}(x) - \text{weight}_{min,t}(x)} \quad (9)$$

The normalized values of constraint are then used to identify the set of essential constraints C_t. For the remaining constraints, we use the GP predictions from the previous iteration to infer the missing observations.

Algorithm Overview

We develop two phases in DMCL. The first expansion phase expands the set of feasible arms while the second optimization phase optimizes the objective in (1). The DMCL is presented in Algorithm 1.

The feasible set at time t is defined as S_t with an inital empty set ϕ. An uncertain set U_t is defined as the complete arm set $\mathcal{X}$ at the beginning. Notably, one of the differences between DMCL and *SafeOpt* lies in the fact that DMCL has no initial safe set, thus allowing the violation of constraints during the exploration phase.

Considering the challenges in computing an accurate estimate of the Lipschitz constant (Wachi and Sui 2020), we expand S_t and reduce U_t based on the confidence intervals of the GP posterior (Chowdhury and Gopalan 2017a) (lines $2-3$ in Algorithm 1). The feasible arm set S_t consists of arms whose lower bounds ℓ_t exceed specific thresholds τ_i, i.e.,

$$S_t = \bigcap_i \bigcup_{x \in S_{t-1}} \{x' \in X \mid \ell_{i,t}(x) \geq \tau_i\} \quad (10)$$

where $\ell_{i,t}(x) := \max(\ell_{i,t-1}(x), \mu_{i,t-1}(x) - \beta_t \sigma_{i,t-1}(x))$. The term β_t is a parameter for the trade-off between exploration (picking points with high uncertainty $\sigma_{i,t-1}(x)$) and exploitation (picking points with high reward $\mu_{i,t-1}(x)$), which will be defined later. The uncertain set U_t is defined as the set of arms whose lower bounds are less than specific thresholds, i.e.,

$$U_t = \bigcap_i \{x \in X \backslash S_t \mid u_{i,t}(x) \geq \tau_i\} \quad (11)$$

where $u_{i,t}(x) := \min(u_{i,t-1}(x), \mu_{i,t-1}(x) + \beta_t \sigma_{i,t-1}(x)$. The confidence interval range is calculated as $w_{i,t}(x) = u_{i,t}(x) - \ell_{i,t}(x)$.

For U_t, whose actions have higher-level uncertainty, includes all possible decisions that can be extended to the feasible arm set S_t. We define an acquisition function to expand the feasible arm set by combining the principle of GP-UCB (Srinivas et al. 2010) and the uncertainty-based selection steps from (Sui et al. 2015) (lines $7-11$ in Algorithm 1). By resorting to the calculation of the accuracy threshold ϵ in (Srinivas et al. 2010), we first obtain $\epsilon_{i,t}$ by calculating the ratio of true function values within the predicted confidence intervals. We choose the next arm from the feasible arm set using GP-UCB when $\epsilon_{i,t} \leq \epsilon$, i.e.,

$$x_t = \text{argmax}_{x \in S_t} \mu_{0,t-1}(x,z) + \beta_t \sigma_{0,t-1}(x,z). \quad (12)$$

Otherwise, we pick the highest predictive uncertain arm to expand S_t:

$$x_t = \text{argmax}_{x \in U_t, i \in [\text{N}]} w_{i,t}(x,z) \quad (13)$$

Next, we incorporate the DCS into our algorithm. We determine which constraints have higher information value by comparing the values of constraint weight $\text{weight}_{i,t}$ and λ, in order to obtain a subset of essential constraints, defined as follows.

$$C_t = \{i \mid i \in N, \text{weight}_{i,t}, \geq \lambda\} \quad (14)$$

which aims to reduce the required sample points while ensuring the performance of preference learning. Then we update the GP posteriors distribution based on observations of the reward function and selected constraint functions (lines $14-17$ in Algorithm 1). The first stage stops under the following condition $U_t = \phi$.

After expanding the potential feasible set, we employ GP-UCB to optimize the reward function within S_t and iteratively search for the best arm within a finite number of iterations T (lines $19-24$ in Algorithm 1).

Theoretical Results

In this section, we show the effectiveness of DMCL by theoretically bounding its cumulative regret and violation.

The effectiveness of DCML crucially relies on the correctness of the confidence intervals of GP predictions. Specifically, to ensure the accuracy of the algorithm, it is required that at each time t, the true values of the aggregation function $f_i(x)$ have a high probability of falling within the confidence intervals $[l_{i,t}(x), u_{i,t}(x)]$, which are determined by the scaling factor β_t. β_t plays an important role in tuning the conservativeness of confidence intervals and its determination has been studied by (Srinivas et al. 2010; Chowdhury and Gopalan 2017b) in the context of the multi-armed bandit. Based on our previous regularity assumptions and Theorem 1 of (Srinivas et al. 2010), we choose β_t as followed:

$$\beta_t = 2b + 300\gamma_t \log^3(t/\delta), \quad (15)$$

where b is the bound on the RKHS norm of the aggregated function $f_i(x)$ and all sample points are corrupted by σ-sub-Gaussian noise, $\delta \in (0,1)$ is a free confidence parameter that represents the tolerable failure probability in DCML. γ_t

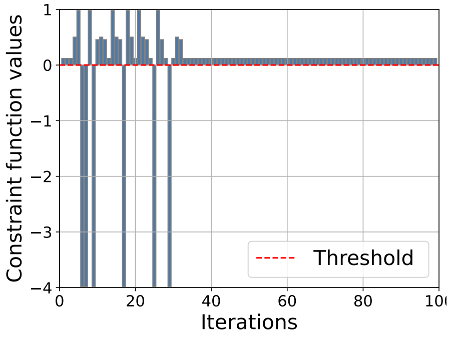

Figure 1: Constraint violation of DMCL in the two-link robot arm system. The red dotted line indicates the constraint threshold.

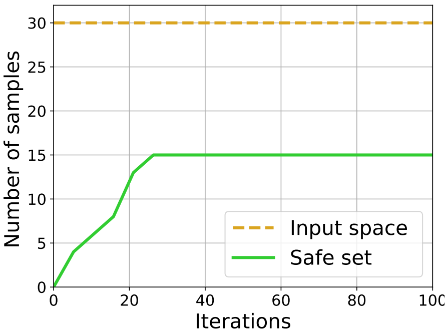

Figure 2: Expansion of the safe set in the two-link robot arm system.

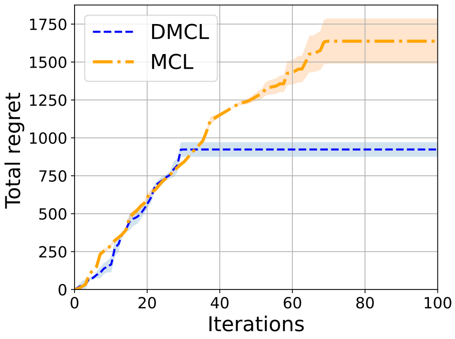

Figure 3: Average cumulative regret by comparing DMCL and MCL in the two-link robot arm environment. The number of arms is fixed at 30.

is the maximum mutual information gain after t-times noisy observation, which is defined as follows by simultaneously considering both the context space and multiple outputs:

$$\gamma_t = \max_{A \subseteq M \times X \times Z, |A| \leq t} I(y_A; f_A), \tag{16}$$

where A is the obtained samples at time t, M refers to the numer of function $N+1$, X denotes the input space, and Z refers to the space of contextual information. The mutual information gain is defined as:

$$I(y_A; f_A) = \frac{1}{2} \log\det |I + \sigma^{-2} K_f|, \tag{17}$$

where $K_f = [k((i,x,z),(i',x',z'))]_{(i,x,z),(i',x',z') \in M \times X \times Z}$. Besides, the DCS module enables us to dynamically choose the most valuable constraints, eliminating the need for obtaining observations for each function in every iteration. We obtain m observations in each iteration, where m $1 \leq m \leq N+1$. The following lemma states that, given β mentioned earlier, the outputs of DCML fall within the confidence interval with high probability.

Lemma 1 *(Confidence Intervals, based on (Srinivas et al. 2010)). Assume that f satisfies $\|f\|_k^2 \leq B$ and that measurements are perturbed by σ-sub-Gaussian noise, pick $\delta \in (0,1)$ and set $\beta_t = 2B + 300\gamma_t \log^3(t/\delta)$ then the following equation holds for all parameters $x \in \mathcal{X}$, function indices $i \in 0, 1, \cdots, n$, and iterations $t \geq 1$ jointly with probability at least $1-\delta$:*

$$|f_i(x_t) - \mu_{i,t-1}(x)| \leq \beta_t \sigma_{i,t-1}(x) \tag{18}$$

where $\mu_{i,t-1}(x)$, $\sigma_{i,t-1}(x)$ are mean and variance of GP posterior distribution.

Proof. This lemma directly follows the Theorem 6 of (Srinivas et al. 2010). The only difference is that we obtain dynamic multiple observations in each iteration, which leads to a fine-tuning of the information capacity γ_t.

Theorem 1 *Fix any $\epsilon > 0$ and $\delta \in (0,1)$, we choose β_t as in Lemma 1, with probability at least $1-\delta$, the following holds:*

$$R_T = \mathcal{O}(\gamma_T \sqrt{T}). \tag{19}$$

$$V_{N,T} = \mathcal{O}(N \gamma_T \sqrt{T}). \tag{20}$$

N refers to the number of constraint functions.

The detailed proof of Theorem 1 is presented in Appendix. We start counting t from the beginning of the exploration stage. The regret bound in Theorem 1 can be found, which is exactly the same as the regret bound in the unconstrained case derived in (Chowdhury and Gopalan 2017a) and the constraint violation bound is similar to the regret bound. Furthermore, based on this theorem, we can derive the accessibility to the optimal solution.

Theorem 2 With Lemma 1, suppose f_i is L_i-Lipschitz-continuous for $i \in [N]$. Fix any $\zeta > 0$ and $\delta \in (0,1)$, we choose β_t as in Lemma 1. T_1 is the time horizon of the optimization stage. During the optimization stage of DMCL, with T being the smallest positive integer, the following satisfies:

$$\frac{8}{\sqrt{T}} \sqrt{B\gamma_T + 150\gamma_T^2 \log^3(t/\delta)} \leq \zeta. \tag{21}$$

With a probability at least $1-\delta$, DCML finds the optimal solution $f(\hat{\boldsymbol{x}}^*) \geq f(\boldsymbol{x}^*) - \zeta$. We start counting t from the beginning of the optimization stage in Theorem 2. We set the time horizon of the optimization stage to T and demonstrate the existence of the optimal solution $\hat{x}^*$ within the safe region during the exploration stage.

Performance Evaluation

In this section, we evaluate the performance of our DCML in two environments. In the first experiment, DCML is used to learn the single-dimensional constraint in a two-link robot arm environment. In the second experiment, DCML learns multiple constraints in a simulated autonomous driving scenario, in order to evaluate its performance in capturing multi-dimensional latent human preferences. We also conducted ablation experiments to analyze the effectiveness of different heuristic modifications. Furthermore, we investigated the impact of using different threshold values for constraint selection in the DMCL framework. Extensive results validate that DCML outperforms existing state-of-the-art approaches.

Two-Link Robot Arm

Experimental Settings The two-link robot arm environment is provided by (Jin et al. 2023), which simulates a

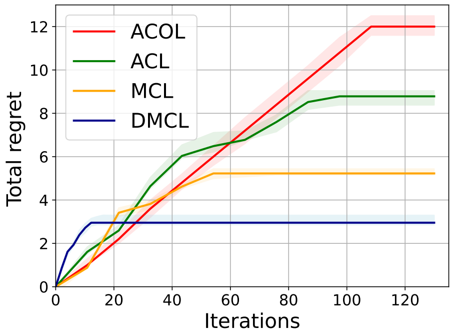

Figure 4: Average cumulative regret obtained from four algorithms. The number of arms is fixed at 10.

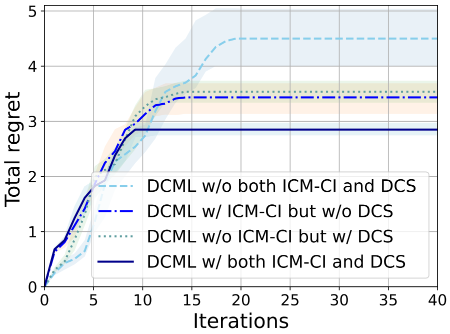

Figure 5: Ablation experiments to evaluate the effect ICM-CI kernel and DCS module in DMCL. The number of arms is fixed at 10.

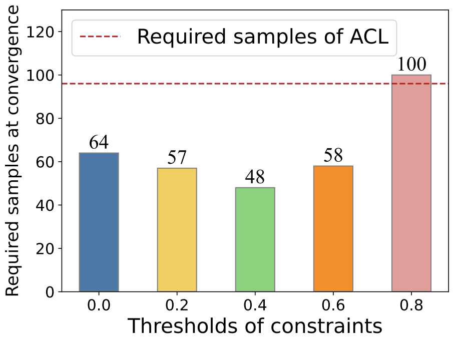

Figure 6: Sample number required by DMCL and ACL (represented by a red dashed line). The number of arms is fixed at 10.

robot learning an objective cost function incrementally from human directional corrections provided via keyboard input. The objective is to teach the robotic arm to reach a target position while avoiding obstacles. We take the cost function as a penalty for the unknown reward function and regard the human directional correction as human preference. The time horizon T is set as 50. Our objective is to identify the optimal arm as accurately and efficiently as possible. We use the ICM kernel with two inputs to jointly model the reward function r and constraint function c, which consists of two components: a Matérn 5/2 kernel (Williams and Rasmussen 2006) that captures shared information among the outputs and a Coregionalize kernel in GPy that constructs the inter-task covariance matrix. We set the constraint threshold τ to 0 and the weight threshold λ to 0 since there is only one constraint.

Results We conduct a set of comparative experiments to compare the convergence performance of DCML and Multi-Constraints Learning (MCL) (see Algorithm 2 in Appendix) and also verify our theoretical results. As a baseline method, MCL employs independent Gaussian kernels for reward and constraint. We first show the constraint value and safe set starting from the exploration stage and then compare the total regret with MCL in the optimization stage.

Figure 1 presents the constraint function values during both the exploration and optimization stage of DMCL. Figure 2 shows the expansion of the safe set S_t as the number of iterations t increases starting from the exploration phase. The starting point of the curve at the y-axis is above 0. As we can see from Figure 3, DCML achieves smaller regret and faster convergence than MCL on average, which means that DCML outperforms MCL.

Autonomous Driving

Experimental Settings We further evaluate DCML with ICM-CI in the autonomous driving environment proposed by (Sadigh et al. 2017).

To reduce computing costs, we set a number of driving policies in the environment and each policy is a combination of multi-dimensional human preferences. The driving policies are then taken as the arms of the CGBAI problem. We set a fixed time horizon $T = 20$ for all experiments. After identifying the arm, the reward and constraints perturbed by noises can be observed. The constraint functions comprise eight dimensions, corresponding to eight-dimensional human preferences on driving behaviors. The reward function only has one dimension, which indicates the step reward of a vehicle. In all experimental settings, both the reward and constraints are initially unknown.

We set T_0 equal to the number of arms starting from a randomly sampled safe action. Additionally, Gaussian noise with a standard deviation of $\sigma = 0.05$ is introduced to the observations. We set the constraint threshold τ_i for each constraint function c_i to be 0 or 1 according to its true value. It is worth noting that the reported iteration refers specifically to the optimization stage of DMCL. We plot the cumulative regret, where the curvature indicates the learning speed of algorithms. When the curvature reaches zero, it means that the corresponding algorithm converges to the optimal solution.

Comparing DMCL to Baselines We compare DMCL approach against MCL, Aggregate Constraint Learning (ACL) (see Algorithm 3 in Appendix), and Adaptive Constraint Learning (ACOL) (Lindner et al. 2022). MCL employs independent Gaussian kernels for each constraint. ACL employs a linear weighting scheme to aggregate all the constraints into a single entity, simplifying the process of observations and predictions to only the aggregated constraint function. In ACOL, both the known reward function and the unknown constraint function are assumed to be linear. It should be mentioned that ACL, MCL, and DMCL are all constrained Gaussian methods, while ACOL is a constrained linear method. We used cumulative regret as our evaluation metric.

Figure 4 compares the performance of DMCL and the baselines in terms of cumulative regret. While all algorithms successfully find the optimal solution, their sample efficiency varies significantly. The ACOL exhibits linear growth of cumulative regret, which requires the maximum number of iterations, resulting in the highest total regret. This demonstrates that constrained Gaussian methods outperform constrained linear methods. ACL performs

Algorithm 1: DCML

```
Input: arm set $\mathcal{X}, i \in \{1, 2, \cdots, n\}$, ICM-CI kernel and a GP prior for aggregation function $f_i$ containing reward function $r$ and constraint functions $\{c_i, \forall i \in [N]\}$, Lipschitz constants $L_i$, safety threshold $\tau_i$, accuracy threshold $\epsilon$, weight threshold $\lambda$.
Output: Optimal constrained arms.
 1: $U_0 \leftarrow \mathcal{X}$.
 2: $S_0 \leftarrow \phi$
 3: $t \leftarrow 1$
 4: while $t \leq T_0$ do
 5:     $S_t \leftarrow \bigcap_i \bigcup_{x \in S_{t-1}} \{x' \in \mathcal{X} \mid \ell_{i,t}(x) - L_i d(x, x') \geq \tau_i\}$
 6:     $U_t \leftarrow \bigcap_i \{x \in X \backslash S_t \mid u_{i,t}(x) - L_i d(x, x') \geq \tau_i\}$
 7:     if $\forall i, \epsilon_{i,t} < \epsilon$ then
 8:         $x_t \leftarrow \text{argmax}_{x \in S_t} \mu_{0,t-1}(x, z) + \beta_t \sigma_{0,t-1}(x, z)$
 9:     else
10:         $x_t \leftarrow \text{argmax}_{x \in U_t, i \in \{1,\ldots,n\}} w_{i,t}(x, z)$
11:     end if
12:     Calculate constraint weights based on (8) and (9)
13:     $C_t \leftarrow \{i \in [N] \mid \text{weight}_{i,t} \geq \lambda\}$
14:     Observe the reward and selected constraints
15:     $y_{0,t} \leftarrow f_0(x_t) + n_{0,t}$
16:     $y_{i,t} \leftarrow f_i(x_t) + n_{i,t}, i \in [N], c_i \in C_t$
17:     Update GP with new observations
18: end while
19: while $t < T$ do
20:     $x_t \leftarrow \text{argmax}_{x,z \in S_t} \mu_{0,t-1}(x, z) + \beta_t \sigma_{0,t-1}(x)$
21:     repeat 12-16
22:     Update GP with new observations
23: end while
24: return $x^* \in \text{argmax}_{x \in S_t} y_{0,t}$
```

better than ACOL since its regret curve exhibits fluctuating growth and ultimately converges faster. MCL further outperforms ACL, which proves that multi-dimensional constraints yield better results compared to one-dimensional constraints. DMCL exhibits the highest sample efficiency, requiring the fewest iterations as anticipated.

Effect of ICM-CI and DCS We further conducted ablation experiments to verify the effectiveness of the two optimization phases in DMCL. In DCML, when using the ICM-CI kernel, accuracy-related contextual information is taken into account. In DCML without the DCS module, all constraints are equally selected for observation. However, with the DCS module, the constraints' weights are computed, and only those surpassing the threshold are selected.

DCML with ICM-CI kernel and DCS module achieved the fastest convergence in terms of the cumulative regret in comparison to the other three baselines. The gap between the results obtained w/o ICM-CI kernel indicates the effectiveness of the new kernel function.

Effect of Constraint Thresholds In DCS, the essential constraints with weights surpassing the threshold will be chosen. To further explore the influence of constraint thresholds on the experimental outcomes, we varied the thresholds and obtained the following results.

Thresholds	Iterations at convergence	Unsafe Actions
0	8	0
0.2	13	0
0.4	22	3
0.6	55	10
0.8	96	20

Table 1: Experimental results averaged over 10 trials with varied thresholds while keeping the number of arms fixed at 10.

From Table 1 we can see, as the threshold increases, the selection possibility of valuable constraints at each iteration decreases, leading to a higher number of iterations required for convergence. Additionally, there is a gradual increase in the occurrence of constraint violations during the iteration process, which means that as the threshold increases, the level of insecurity also increases.

However, the total number of constraint samples required for convergence does not increase with the number of iterations. Figure 6 illustrates that when the threshold is set to 0.4, the sample count reaches its minimum, with an average selection of 2 constraints per iteration. When the threshold is set to 0, all constraints are equally selected, resulting in the selection of 8 constraints in each iteration. Therefore, the total sample count is directly proportional to the number of iterations, with a ratio of 8:1. However, it is not the optimal outcome, suggesting that not all constraints contribute equally to the selection process, thus providing further evidence of the effectiveness of the DCS module. ACL aggregates the eight constraints using a weighted sum approach. It only needs to observe the aggregated constraint value, resulting in a number of iterations equal to the required number of samples. Setting the threshold to 0.8 is the only scenario where the required number of samples exceeds that of ACL. This observation suggests that setting a large threshold can lead to increased uncertainty in the system.

Conclusion

In this work, we study human-machine complementarity in hybrid systems by formulating a CGBAI problem and then propose a new DMCL algorithm. The key insight we aim to convey includes two aspects: 1), Although human preferences affect machines' decisions in a time-varying and latent fashion, we can capture their temporal correlation and combined impact from the interactions progressively, which can ensure the learned preferences robust to different system objectives; 2), Human preferences inherently display multidimensionality, yet this trait is not always essential for achieving complementarity. In addition, we contribute to the existing works related to the understanding of human-machine complementarity by theoretically and empirically verifying the effectiveness of the Bayesian learning framework in addressing sequential decision-making problems within hybrid systems.

References

Alvarez, M. A.; Rosasco, L.; Lawrence, N. D.; et al. 2012. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3): 195–266.

Amani, S.; Alizadeh, M.; and Thrampoulidis, C. 2020. Regret bound for safe gaussian process bandit optimization. In *Learning for Dynamics and Control*, 158–159. PMLR.

Berkenkamp, F.; Krause, A.; and Schoellig, A. P. 2021. Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *Machine Learning*, 1–35.

Bonilla, E. V.; Chai, K. M. A.; and Williams, C. K. I. 2007. Multi-task Gaussian Process Prediction. In *NIPS*.

Brochu, E.; Cora, V. M.; and De Freitas, N. 2010. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *CoRR, abs/1012.2599*.

Chowdhury, S. R.; and Gopalan, A. 2017a. On Kernelized Multi-Armed Bandits. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, 844–853. JMLR.org.

Chowdhury, S. R.; and Gopalan, A. 2017b. On Kernelized Multi-armed Bandits. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 844–853. PMLR.

Elmalaki, S. 2021. FaiR-IoT: Fairness-Aware Human-in-the-Loop Reinforcement Learning for Harnessing Human Variability in Personalized IoT. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, IoTDI '21, 119–132. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383547.

Jin, W.; Murphey, T. D.; Lu, Z.; and Mou, S. 2023. Learning From Human Directional Corrections. *IEEE Transactions on Robotics*, 39(1): 625–644.

Kelly, M.; Kumar, A.; Smyth, P.; and Steyvers, M. 2023. Capturing Humans' Mental Models of AI: An Item Response Theory Approach. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, 1723–1734.

Kerrigan, G.; Smyth, P.; and Steyvers, M. 2021. Combining human predictions with model probabilities via confusion matrices and calibration. *Advances in Neural Information Processing Systems*, 34: 4421–4434.

Krause, A.; and Ong, C. 2011. Contextual gaussian process bandit optimization. *Advances in neural information processing systems*, 24.

Lindner, D.; Tschiatschek, S.; Hofmann, K.; and Krause, A. 2022. Interactively Learning Preference Constraints in Linear Bandits. In *International Conference on Machine Learning*.

Liu, Z.; Wang, J.; Gong, S.; Lu, H.; and Tao, D. 2019. Deep reinforcement active learning for human-in-the-loop person re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6122–6131.

Mao, H.; Venkatakrishnan, S. B.; Schwarzkopf, M.; and Alizadeh, M. 2018. Variance reduction for reinforcement learning in input-driven environments. *arXiv preprint arXiv:1807.02264*.

Patel, B. N.; Rosenberg, L.; Willcox, G.; Baltaxe, D.; Lyons, M.; Irvin, J.; Rajpurkar, P.; Amrhein, T.; Gupta, R.; Halabi, S.; et al. 2019. Human–machine partnership with artificial intelligence for chest radiograph diagnosis. *NPJ digital medicine*, 2(1): 111.

Sadigh, D.; Dragan, A. D.; Sastry, S. S.; and Seshia, S. A. 2017. Active Preference-Based Learning of Reward Functions. In *Robotics: Science and Systems*.

Srinivas, N.; Krause, A.; Kakade, S.; and Seeger, M. 2010. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, 1015–1022. Madison, WI, USA: Omnipress. ISBN 9781605589077.

Steyvers, M.; Tejeda, H.; Kerrigan, G.; and Smyth, P. 2022. Bayesian modeling of human–AI complementarity. *Proceedings of the National Academy of Sciences*, 119(11): e2111547119.

Sui, Y.; Gotovos, A.; Burdick, J. W.; and Krause, A. 2015. Safe Exploration for Optimization with Gaussian Processes. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, 997–1005. JMLR.org.

Sui, Y.; Zhuang, V.; Burdick, J.; and Yue, Y. 2018. Stagewise Safe Bayesian Optimization with Gaussian Processes. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 4781–4789. PMLR.

Ustalov, D.; Fedorova, N.; and Pavlichenko, N. 2022. Improving Recommender Systems with Human-in-the-Loop. In *Proceedings of the 16th ACM Conference on Recommender Systems*, 708–709.

Wachi, A.; and Sui, Y. 2020. Safe Reinforcement Learning in Constrained Markov Decision Processes. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Williams, C. K.; and Rasmussen, C. E. 2006. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

Wu, Y.-H.; Charoenphakdee, N.; Bao, H.; Tangkaratt, V.; and Sugiyama, M. 2019. Imitation Learning from Imperfect Demonstration. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 6818–6827. PMLR.

Xu, W.; Jiang, Y.; Svetozarevic, B.; and Jones, C. 2023. Constrained efficient global optimization of expensive black-box functions. In *International Conference on Machine Learning*, 38485–38498. PMLR.

Zhou, X.; and Ji, B. 2022. On kernelized multi-armed bandits with constraints. *Advances in Neural Information Processing Systems*, 35: 14–26.

A. Proofs

Corollary 1 *Under the assumptions above. We choose β_t as in Lemma 1, with probability at least $1-\delta$, for all $x \in \mathcal{X}$ and $1 \le t \le T$, the following holds:*

$$f_i(x) \in [\ell_{i,t}(x), u_{i,t}(x)]$$

Proof. Based on the result of Lemma 1, we have

$$f_i(x) \in [\mu_{i,t-1}(x) - \beta_t \sigma_{i,t-1}(x), \mu_{i,t-1}(x) + \beta_t \sigma_{i,t-1}(x)]$$

According to the definition

$$\ell_{i,t}(x) := max(\ell_{i,t-1}(x), \mu_{i,t-1}(x) - \beta_t \sigma_{i,t-1}(x)),$$

we can verify the correctness of 1 by demonstrating $f_i(x) \in [\ell_{i,t-1}(x), u_{i,t-1}(x)]$. By following the same reasoning process, we only need $f_i(x) \in [\ell_{i,0}(x), u_{i,0}(x)]$ to prove this corollary, which is clearly based on Lemma 1.

Lemma 2 *Under the assumptions above, with probability at least $1-\delta$, for all $x \in \mathcal{X}$ and $1 \le t \le T$, the following holds:*

$$R_t \le 2\beta_t \sigma_{0,t-1}(x)$$

Proof. By the definition, stepwise regret is derived as:

$$\begin{aligned} R_t &= f_0\left(x^*\right) - f_0\left(x_t\right) \\ &\le \mu_{0,t-1}\left(x_t\right) + \beta_t \sigma_{0,t-1}\left(x_t\right) - f_0\left(x_t\right) \\ &\le \mu_{0,t-1}\left(x_t\right) + \beta_t \sigma_{0,t-1}\left(x_t\right) - \left(\mu_{0,t-1}\left(x_t\right) - \beta_t \sigma_{0,t-1}\left(x_t\right)\right) \\ &= 2\beta_t \sigma_{0,t-1}\left(x_t\right). \end{aligned}$$

Lemma 3 *Under the assumptions above. With probability at least $1-\delta$, for all $x \in \mathcal{X}$,$i \in [N]$ and $1 \le t \le T$, the following holds:*

$$v_{i,t} \le 2\beta_t \sigma_{i,t-1}(x), i \in [N]$$

Proof. By the definition, stepwise violation is derived as:

$$\begin{aligned} v_{i,t} &= \left[\tau_i - f_i\left(x_t\right)\right]^+ \\ &= \left[\tau_i - \ell_{i,t}(x) + \ell_{i,t}(x) - f_i\left(x_t\right)\right]^+ \\ &\le \left[\tau_i - \ell_{i,t}(x)\right]^+ + \left[\ell_{i,t}(x) - f_i\left(x_t\right)\right]^+ \\ &\le \left[\tau_i - \ell_{i,t}(x)\right]^+ + \left[u_{i,t}(x) - f_i\left(x_t\right)\right]^+ \\ &\le \left[\tau_i - \ell_{i,t}(x)\right]^+ + \left[u_{i,t}(x) - \tau_i\right]^+ \\ &= \left[u_{i,t}(x) - \ell_{i,t}(x)\right]^+ \\ &= 2\beta_t \sigma_{i,t-1}\left(x_t\right). \end{aligned}$$

Lemma 4 *(Cumulative variance bound, based on (Chowdhury and Gopalan 2017a)) Suppose that we select a sample point x_t in each iteration t. After T iterations, for all $i \in \{0\} \cup [N]$,we have*

$$\Sigma_{t=1}^{T} \sigma_{i,t-1}\left(x_t\right) \le \sqrt{4(T+2)\gamma_T}$$

where γ_T is the maximum information gain from f_i after T iterations.

Proof. This lemma directly follows the Lemma 4 of (Chowdhury and Gopalan 2017b)

Proof of Theorem 1:

$$\begin{aligned} R_T &= \Sigma_{t=1}^{T} R_t \\ &\le 2\beta_t \Sigma_{t=1}^{T} \sigma_{0,t-1}\left(x_t\right) \\ &\le 2\beta_T \sqrt{4(T+2)\gamma_T} \\ &\le 4\sqrt{2B + 300\gamma_T \log^3(t/\delta)}\sqrt{(T+2)\gamma_T}. \\ &\le 4\sqrt{2B + 300\gamma_T \log^3(t/\delta)}\sqrt{2T\gamma_T} \\ &= 8\sqrt{BT\gamma_T + 150T\gamma_T^2 \log^3(t/\delta)} \\ &= \mathcal{O}(\gamma_T \sqrt{T}) \end{aligned}$$

$$\begin{aligned} V_{N,T} &= \sum_{i=1}^{N} \sum_{t=1}^{T} v_{i,t} \\ &= \sum_{i=1}^{N} \sum_{t=1}^{T} \left[\tau_i - f_i\left(x_t\right)\right]^+ \\ &\le \sum_{i=1}^{N} \sum_{t=1}^{T} 2\beta_T \sigma_{i,t-1}\left(x_t\right) \\ &= 2\beta_T \sum_{i=1}^{N} \sqrt{4(T+2)\gamma_T} \\ &= \sum_{i=1}^{N} 8\sqrt{BT\gamma_T + 150T\gamma_T^2 \log^3(t/\delta)} \\ &= \mathcal{O}(N\gamma_T \sqrt{T}) \end{aligned}$$

Proof of Theorem 2: Base on Theorem 1, with a probability of at least $1-\delta$, the average regret $\bar{R}_T$ adheres to the following relationship:

$$\bar{R}_T = \frac{R_T}{T} \le \frac{8}{\sqrt{T}} \sqrt{B\gamma_T + 150\gamma_T^2 \log^3(t/\delta)}.$$

Let T be the smallest positive integer that satisfies the following condition:

$$\frac{8}{\sqrt{T}} \sqrt{B\gamma_T + 150\gamma_T^2 \log^3(t/\delta)} \le \zeta.$$

Consequently, we can directly obtain $\bar{R}_T = R_T/T \le \zeta$. This means that the ζ-optimal performance can be achieved by the DMCL algorithm with at least $1-\delta$, denoted by $f\left(\hat{\boldsymbol{x}}^*\right) \ge f\left(\boldsymbol{x}^*\right) - \zeta$.

B. Experimental Details

Two-Link Robot Arm System We extend the Two-Link Robot Arm System proposed by (Jin et al. 2023) and we provide a brief description of the dynamics and features of the environment. The dynamics of the two-link robot arm system use inertia matrix, Coriolis and centrifugal term. The state and control input are $\boldsymbol{x} = [\boldsymbol{q}, \dot{\boldsymbol{q}}]^\top \in \mathbb{R}^4$ and $\boldsymbol{u} = \boldsymbol{\tau} \in \mathbb{R}^2$, respectively. The initial condition of the robot arm is set as $\boldsymbol{x}_0 = [-\frac{\pi}{2}, 0, 0, 0]^\top$. The robot cost function has the following parameterized form:

$$J\left(\boldsymbol{u}_{0:T}, \boldsymbol{\theta}\right) = \sum_{t=0}^{T} \boldsymbol{\theta}^\top \boldsymbol{\phi}\left(\boldsymbol{x}_t, \boldsymbol{u}_t\right) + h\left(\boldsymbol{x}_{T+1}\right)$$

where $\phi : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^r$ is a vector of the predefined features; $\boldsymbol{\theta} \in \mathbb{R}^r$ is a vector of weights, which are tunable; and $h\left(\boldsymbol{x}_{T+1}\right)$ is the final cost on the robot final state $\boldsymbol{x}_{T+1}$, $h\left(\boldsymbol{x}_{T+1}\right) = 100\left(\left(q_1 - \frac{\pi}{2}\right)^2 + q_2^2 + \dot{q}_1^2 + \dot{q}_2^2\right)$. In the cost function , the feature and weight vectors are set to

$$\phi = \left[q_1^2, q_1, q_2^2, q_2, \|u\|^2\right]^\top \in \mathbb{R}^5,$$
$$\theta = \left[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5\right]^\top \in \mathbb{R}^5$$

respectively and they are known before. The robot arm aims to reach and stop at the pose of $\boldsymbol{q} = \left[\frac{\pi}{2}, 0\right]^\top$ in discrete-time horizon $T = 50$. The dynamics is discretized using the Euler method with a time interval $\Delta = 0.2\,\mathrm{s}$ The environment dynamics and the initial condition are given by

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}\left(\boldsymbol{x}_t, \boldsymbol{u}_t\right), \text{ with } \boldsymbol{x}_0$$

where $\boldsymbol{x}_t \in \mathbb{R}^n$ is the robot state, $\boldsymbol{u}_t \in \mathbb{R}^m$ is the control input, $\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ is differentiable, and $t = 1, 2, \ldots$ is the time step.

To adapt the environment to our problem setting, we replace the constraints with punitive rewards as an easy-to evaluate aspect, while incorporating human directional corrections through keyboard input as the expensive-to-evaluate human preferences for learning. In each For a fixed choice of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, In each game, the robot plans a sequence of inputs $\boldsymbol{u}_{0:T}$ over time horizon T by (locally) optimizing the cost function subject to the environment dynamics, producing a trajectory and we utilize the four arrow keys on the keyboard as the interface for human intervention in controlling the robot's motion trajectory. By repeating this process thirty times, we obtain thirty trajectories as the arms in our problem setting.

C. Alternative Algorithms

For the sake of completeness, we provide the pseudocode of both MCL and ACL.

Algorithm 2: MCL

Input: arm set $X, i \in \{1, 2, \cdots, n\}$, GP prior for reward function r and GP priors for constraint functions c_i, Lipschitz constants L_i for c_i , safety threshold τ_i, accuracy threshold ϵ, weight threshold λ.
Output: Optimal constrained arm.

```
 1: $U_0 \leftarrow X$.
 2: $S_0 \leftarrow \phi$
 3: $t \leftarrow 1$
 4: while $t \leq T_0$ do
 5:     $S_t \leftarrow \bigcap_i \bigcup_{x \in S_{t-1}} \{x' \in X \mid \ell_{i,t}(x) - L_i d(x, x') \geq \tau_i\}$
 6:     $U_t \leftarrow \bigcap_i \{x \in X \backslash S_t \mid u_{i,t}(x) - L_i d(x, x') \geq \tau_i\}$
 7:     if $\forall i, \epsilon_t^i < \epsilon$ then
 8:         $x_t \leftarrow argmax_{x \in S_t} \mu_{0,t-1}(x) + \beta_t \sigma_{0,t-1}(x)$
 9:     else
10:         $x_t \leftarrow argmax_{x \in U_t, i \in \{1,\ldots,n\}} w_{i,t}(x)$
11:     end if
12:     Observe all constraints and reward value
13:     $y_{0,t} \leftarrow r(x_t) + n_{0,t}$
14:     $y_{i,t} \leftarrow c_i(x_t) + n_{i,t}$
15:     Update GP with new data
16: end while
17: while $t < T$ do
18:     $x_t \leftarrow argmax_{x \in S_t} \mu_{0,t-1}(x) + \beta_t \sigma_{0,t-1}(x)$
19:     repeat 12-15
20:     Update GP with new data
21: end while
22: return $x^* \in argmax_{x \in S_t} y_{r,t}$
```

Algorithm 3: ACL

Input: arm set X, GP prior for reward function r and GP prior for constraint function c, Lipschitz constant L for c , safety threshold τ, accuracy threshold ϵ, weight threshold λ.
Output: Optimal constrained arm.

```
 1: $U_0 \leftarrow X$.
 2: $S_0 \leftarrow \phi$
 3: $t \leftarrow 1$
 4: while $t \leq T_0$ do
 5:     $S_t \leftarrow \bigcup_{x \in S_{t-1}} \{x' \in X \mid \ell_{c,t}(x) - L d(x, x') \geq \tau\}$
 6:     $U_t \leftarrow \{x \in X \backslash S_t \mid u_{c,t}(x) - L d(x, x') \geq \tau\}$
 7:     if $\forall i, \epsilon_t^i < \epsilon$ then
 8:         $x_t \leftarrow argmax_{x \in S_t} \mu_{r,t-1}(x) + \beta_t \sigma_{r,t-1}(x)$
 9:     else
10:         $x_t \leftarrow argmax_{x \in U_t} w_{c,t}(x)$
11:     end if
12:     Observe constraint and reward value
13:     $y_{r,t} \leftarrow r(x_t) + n_{r,t}$
14:     $y_{c,t} \leftarrow c(x_t) + n_{c,t}$
15:     Update GP with new data
16: end while
17: while $t < T$ do
18:     $x_t \leftarrow argmax_{x \in S_t} \mu_{r,t-1}(x) + \beta_t \sigma_{r,t-1}(x)$
19:     repeat 12-15
20:     Update GP with new data
21: end while
22: return $x^* \in argmax_{x \in S_t} y_{r,t}$
```