Learning Submodular Objectives for Team Environmental Monitoring

Nils Wilde, Armin Sadeghi, and Stephen L. Smith

Abstract—In this paper, we study the well-known team orienteering problem where a fleet of robots collects rewards by visiting locations. Usually, the rewards are assumed to be known to the robots; however, in applications such as environmental monitoring or scene reconstruction, the rewards are often subjective and specifying them is challenging. We propose a framework to learn the unknown preferences of the user by presenting alternative solutions to them, and the user provides a ranking on the proposed alternative solutions. We consider the two cases for the user: 1) a deterministic user which provides the optimal ranking for the alternative solutions, and 2) a noisy user which provides the optimal ranking according to an unknown probability distribution. For the deterministic user we propose a framework to minimize a bound on the maximum deviation from the optimal solution, namely regret. We adapt the approach to capture the noisy user and minimize the expected regret. Finally, we demonstrate the importance of learning user preferences and the performance of the proposed methods in an extensive set of experimental results using real world datasets for environmental monitoring problems.

Index Terms—Incremental Learning, Multi-Robot Systems, Environment Monitoring and Management

I. INTRODUCTION

Autonomous multi-robots systems find wide-spread acceptance in an increasing number of applications such as persistent monitoring, environmental data collection, shared autonomy and scene reconstruction. A key challenge remains the design of frameworks that allow users who are not robotic experts to deploy them effectively and efficiently.

We study a generalized version of the well known Team-Orienteering Problem (TOP) [1] where a fleet of robot has to visit multiple locations in the environment. Upon visit, the respective robot collects a reward and the objective is to maximize the total reward collected by the fleet, subject to constraints on the robots' maximum travel distance. Multiple variants have been studied, including uncertainty [2], [3], and complex reward functions modelling correlations [4] and diminishing returns [5].

In some applications, such as servicing tasks or delivery, the reward is directly given, e.g., as a monetary value. However, in other applications the reward can be difficult to quantify

Fig. 1: Tours for a fleet of two robots for different reward functions with high (green) and low (red) rewards assigned to regions. (a) shows a scenario the robots are not aware of the user preferences over the regions and prioritize the regions equally. In (b) the robots have learned an estimate of the user preference over the regions and identified that re-visiting the green region is more valuable.

and might be user dependent. For example, in environmental monitoring, scientists may have differing opinions on the importance of gaining information in a certain region. Often the user can indicate regions of interest that the robots should visit. Yet, defining numerical values for a reward function to prioritize between regions is challenging. This is further enhanced when the reward exhibits a diminishing return property: Additional visits of the same region have decreasing additional value. Thus, defining reward functions becomes impractical, especially when the user is not a robotics expert.

In human-robot interaction (HRI) the problem of defining reward functions is known as *reward design*. To reduce the complexity and thus enable a broader range of users to deploy autonomous robots, researchers have studied different frameworks for *reward learning* [6]–[17]. In contrast to designing parameters of a reward function, users interact with the robot via modalities such as demonstrations, corrections, critique, or choice feedback.

We apply learning from choice to enable users to specify complex submodular reward functions for GTOP and present new solution techniques that are able to handle the high number of dimensions often encountered in these problems. In our framework the robot fleet is given a set of areas of interest. Over multiple iterations, the user is (virtually) presented with two different sets of tours for the robot fleet; they then choose the preferred option. Using a finite set of submodular basis functions, the user's choice allows the robot fleet to estimate the user reward function. Figure 1 illustrates an example for environmental data collection. The regions of interest are protected areas along a coastline. Without designing or learning a reward function, the robots prioritize them equally (a). Learning from choice feedback allows the robot fleet to

identify which regions are most relevant to the user and finds better tours (b).

Contributions: In this paper we make the following contributions: (1) We design submodular basis functions to describe rewards for the generalized team orienteering problem. (2) We propose a novel heuristic policy for active preference learning that can handle a high number of basis functions. (3) To handle uncertainty in the user feedback, we present a novel framework that casts this probabilistic problem to a distribution over instances with noiseless feedback, allowing for efficient learning under uncertainty. (4) Finally, we demonstrate the practicality of the approach in simulation using real-world locations for environmental data collection.

Related Work: We address the challenge of defining reward functions for generalized team-orienteering problems and propose an interactive learning framework. Similarly, researchers in HRI study the design of interactive frameworks that allow inexperienced users to define reward functions for autonomous robots in a wide range of applications. Since classical approaches such as learning from demonstrations are not always suitable, alternative modes of interaction including corrections, proxy rewards, critique, and choice have been developed [6]. This work is based on *learning from choice* (sometimes also referred to as active preference learning), where a user iteratively chooses between two presented options [7], [9]– [11], [16], [18]. Similar to existing work, we pose the problem as learning weights in a linear reward function. We make novel contributions to address challenges arising from the high dimensionality often found in multi-robot problems. Existing approaches usually rely on sampling potential solutions as well as weights for the reward function [7], [9]. We study how the min-max regret technique from [11] can be extended so it can be used without any samples in a noiseless setting. Further, to handle noisy user feedback we propose a method to cast such feedback to multiple noiseless instances and solve the problem on them.

We focus on a generalized version of the team orienteering problem (TOP) [1], which is NP-hard. Using basis functions we consider variances of TOP where reward functions can be correlated between vertices, as well as have a diminishing return, i.e., are submodular [19]. The authors of [4] study OPs with correlated rewards and give a Mixed-Integer-Quadratic-Program solution, which can also be applied to the multi-robot case. For single-robot OP with submodular rewards, the authors of [5] provide a constant factor approximation algorithm. The authors of [20] propose an approximation algorithm for TOP by sequentially solving single OPs for each robot. We combine the latter two techniques to obtain a constant factor approximation for submodular TOP.

Stochastic variants of orienteering problems include uncertainty on edge weights [3], [21], time to service a location [22], and rewards [2]. Similar to the latter case, we study the problem where the robot fleet does not know the rewards. In [23] the rewards of an orienteering problem are dynamic and depend on measurements taken at previous locations. In [2] robots learn the reward function by iteratively executing tours and use these observations to improve future tours. In contrast, our framework allows robots to learn the reward function by querying a user; the user then does not assign a reward to a single set of tours but instead chooses the preferred set of tours among two presented options. Another difference to [2] is that our framework can be used as an offline method, where the user is shown tours virtually allowing robots to learn the reward function prior to execution.

Potential applications of the proposed framework include persistent monitoring [24], [25], environmental data collection [23], and scene reconstruction [26]. The authors of [27], [28] propose an interactive framework for marine data collection: Users define desired targets for observation, the robot then proposes alternatives based on additional information about risks in the environment. Similar to our work, users then choose between different options. This allows the robot to learn the user's utility function trading off reward and risk.

II. PROBLEM FORMULATION

Consider a set of m robots collecting information in an environment represented by a graph $G = (V, E, l)$. The set V is the set of vertices, E is the set of edges between the vertices, and $l : E \to \mathbb{R}_{\geq 0}$ assigns costs to the edges of the graph. A tour T is a sequence of vertices $\langle v_0, v_1, \ldots, v_n, v_0 \rangle$. Given a depot location $s \in V$, a tour starts at s, i.e., $v_0 = s$. The reward function $R : 2^V \to \mathbb{R}_+$ assigns a reward to each set of vertices, i.e., the reward of visiting the vertices in a tour T denoted by $V(T)$. With a slight abuse of notation we write the reward $R(V(T))$ simply as $R(T)$.

Problem 1 (Orienteering Problem)**.** Given a graph $G = (V, E, l)$, a reward function $R : 2^V \to \mathbb{R}_{\geq 0}$ and a positive number B, find a tour T of length at most B that maximizes the reward collected $R(T)$.

We are interested in monotone, normalized submodular reward functions [19]. Such a function has the following properties: i) $R(\emptyset) = 0$, ii) $R(A) \leq R(B)$ for every $A \subseteq B \subseteq V$, and iii) $R(A \cup \{v\}) - R(A) \geq R(B \cup \{v\}) - R(B)$ for every $A \subseteq B \subseteq V$ and for every $v \in V$. Now we introduce a generalization of Problem 1 where there are m heterogeneous robots maximizing a submodular reward function.

Problem 2 (Generalized Team Orienteering Problem (GTOP))**.** Consider a graph $G = (V, E, l)$, a fleet of m robots with travel budgets $B_1, \ldots, B_m$, a partition of the vertices into m subsets $V_1, \ldots, V_m$, and a submodular reward function $R : 2^V \to \mathbb{R}_{\geq 0}$. Find a set of m tours $\mathcal{T} = \{T_1, \ldots, T_m\}$ maximizing the total reward collected subject to the constraints that T_i for all $i \in \{1, \ldots, m\}$ only visits vertices in V_i and has length at most B_i, i.e.,

$$\max_{\mathcal{T}} \quad R(\cup_{i=1}^m V(T_i))$$
$$\text{subject to:} \quad \ell(T_i) \leq B_i, V(T_i) \subseteq V_i \ \forall i \in \{1, \ldots, m\} \tag{1}$$

Remark 1 (Comments on Problem Formulation). The set V_i contains a vertex for each location that robot i can visit. Each vertex is contained in just one set. Thus, if multiple robots can visit the same location, then each has a corresponding vertex in their set V_i. The advantage of this representation is that each vertex encodes both the location and the robot performing a

visit, and thus a submodular function can be defined directly over sets of vertices. This is in contrast to other formulations of the submodular team orienteering problem [3], [29] where the submodular function is defined over the set of all tours/paths, which is exponential in the number of vertices. •

GTOP with Unknown Reward Function: In this work we consider the case where the reward function R is unknown to the robot. We denote the hidden optimal reward function as R^* and the corresponding GTOP solution $\mathcal{T}^*$. Further, let $\hat{R}$ be a robot's estimate of the reward function and $\hat{T}$ the corresponding optimal tour; we are interested in finding an estimate $\hat{R}$ of the reward function with corresponding optimal tours $\hat{\mathcal{T}} = \{\hat{T}_1, \ldots, \hat{T}_m\}$ that solves

$$\max_{\hat{\mathcal{T}}} \quad R^*(\hat{\mathcal{T}}) \quad \text{subject to: } \ell(\hat{T}_i) \leq B_i \text{ for all } \hat{T}_i \in \hat{\mathcal{T}}. \quad (2)$$

We notice that this is an ill-posed problem as the reward function R^* is not available to the robot. However, we consider a framework where the robot iteratively interacts with the user, allowing it to make observations about the user's hidden reward function. This is known as *reward learning*, where the robot presents the user with one or multiple possible solutions to it's task and then obtains feedback in the form of corrections, choice, labels, or others [6]. We define a query as a set of possible solutions for the GTOP $Q = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_l\}$. Let $f(R)$ be some prior belief over the set of all possible reward functions $\mathcal{R}$. Given feedback U, the robot can compute a posterior $f(R|(Q, U))$. We can express the expected outcome with respect to the prior $\mathbb{E}_{U \sim f(R)}[R|(Q, U)]$. This framework allows us to state our problem as an adaptive stochastic optimization problem:

Problem 3 (Learning GTOP Rewards). Given $G = (V, E, l)$, a hidden reward function R^*, a fleet of m robots with travel budgets $B_1, \ldots, B_m$; find a sequence of K queries $(Q_1, Q_2, \ldots, Q_K)$ such that the expected estimated reward function $\hat{R} = \mathbb{E}_{U_k \sim f(R)}[R|(Q_1, U_1), \ldots, (Q_K, U_K)]$ and the corresponding sets of m tours $\hat{\mathcal{T}}$ solves (2).

III. TEAM ORIENTEERING PROBLEM WITH SUBMODULAR BASIS FUNCTIONS

In this section, we present a linear approximation of a submodular reward function, then we propose an approximation algorithm for the GTOP for the linearized reward function.

A. Basis functions

We consider the reward function $R(\mathcal{T})$ to be be composed of a set of basis functions $r_1, \ldots, r_n : 2^V \to \mathbb{R}_{\geq 0}$. Given tours $\mathcal{T}$, the reward function is then a weighted sum of the basis functions $R(\mathcal{T}, \boldsymbol{w}) = \sum_{i=1}^n w_i r_i(\mathcal{T})$.

Similar approaches are commonly used in reward learning problems [6], [7], [9]–[11], where basis functions are usually referred to as *features*. Given that r_i depend only on the vertices, we can assume without loss of generality that each basis function is characterized by a subset $W_i \subseteq V$. That is, for any W_i, let $\psi_i(\mathcal{T})$ be a count of how many vertices of the tours $\mathcal{T}$ lie in W_i, then $r_i(\mathcal{T})$ is a functional of $\psi_i(\mathcal{T})$.

The subsets $W_1, \ldots, W_n$ can reflect a spatial relation between vertices, i.e., describe neighborhoods, but can also express other features, such as grouping all vertices where the robots can make certain observations. The basic case $r_i(\mathcal{T}) = \psi_i(\mathcal{T})$ is a modular function describing the number of times subset W_i is visited. However, many real-world problems exhibit a diminishing return property [30], [31]. In order for r_i to be growing *submodularly* with ψ_i, we choose

$$r_i(\mathcal{T}) = \sum_{\alpha=1}^{\psi_i(\mathcal{T})} \gamma^{(\alpha-1)}, \quad (3)$$

where $\gamma \in (0, 1]$. If $\gamma = 1$ we recover the modular case; in the other extreme that $\gamma \to 0$, visiting W_i more than once effectively does not yield a larger reward than the first visit.

Using these basis functions, the problem of learning the user reward function R^* becomes one of learning the weights $\boldsymbol{w}^* = (w_1^*, \ldots, w_n^*)$, i.e., the importance of each basis function, as well as the decay parameters $\gamma_1^*, \ldots, \gamma_n^*$ describing the diminishing return. Unfortunately, the proposed reward function is only linear in the weights $\boldsymbol{w}$, but not in the decays. Therefore, we assume that γ comes from a discrete set Γ. For each subset W_i we define $|\Gamma|$ basis functions $r_{i,j}$ for $j = 1, \ldots, |\Gamma|$, using the different values γ_j. Using this discretization the overall reward function becomes

$$R(\mathcal{T}, \boldsymbol{w}) = \sum_{i=1}^n \sum_{j=1}^{|\Gamma|} w_{i,j} r_{i,j}(\mathcal{T}). \quad (4)$$

For a sparse notation let $\boldsymbol{\phi} = [r_1, r_2, \ldots]$, allowing us to write $R(\mathcal{T}, \boldsymbol{w}) = \boldsymbol{\phi}(\mathcal{T}) \cdot \boldsymbol{w}$. Further, for any given weight $\boldsymbol{w}$, let $\mathcal{T}(\boldsymbol{w})$ denote the set of tours maximizing the reward, i.e., $\mathcal{T}(\boldsymbol{w}) = \arg\max_{\mathcal{T}} R(\mathcal{T}, \boldsymbol{w})$. Consequently, $\boldsymbol{\phi}(\mathcal{T})$ denotes the features of the tour $\mathcal{T}$.

Observation 1 (Submodularity). The reward function $R(\mathcal{T}, \boldsymbol{w})$ proposed in Equation (4) is a normalized, monotone and submodular set function.

Proof. Since $\psi_i(\emptyset) = 0$ for all i, we have $R(\emptyset, \boldsymbol{w}) = 0$; hence, the function is normalized. Further $V(\mathcal{T}') \supseteq V(\mathcal{T})$ implies $\psi_i(\mathcal{T}') \geq \psi_i(\mathcal{T})$; adding an additional vertex can only increase the vertex count of any set W_i. Hence, $\sum_{\alpha=1}^{\psi_i(\mathcal{T}')} \gamma_j^{(\alpha-1)} \geq \sum_{\alpha=1}^{\psi_i(\mathcal{T})} \gamma_j^{(\alpha-1)}$ for any $V(\mathcal{T}') \supseteq V(\mathcal{T})$ and any $\gamma_j \in (0, 1]$, making R monotonically increasing. Finally, consider any vertex v and two sets of tours $\mathcal{T}$ and $\mathcal{T}'$ where $V(\mathcal{T}') \supseteq V(\mathcal{T})$. Then $\psi_i(\mathcal{T}') \geq \psi_i(\mathcal{T})$ for all i. If $v \in W_i$, the marginal gain is $\Delta(\mathcal{T}, v) = r_{i,j}(\mathcal{T} \cup v) - r_{i,j}(\mathcal{T}) = \gamma_j^{\psi_i(\mathcal{T})+1}$ and $\Delta(\mathcal{T}', v) = r_{i,j}(\mathcal{T}' \cup v) - r_{i,j}(\mathcal{T}') = \gamma_j^{\psi_i(\mathcal{T}')+1}$. Since $\gamma_j \in (0, 1]$ and $\psi_i(\mathcal{T}') \geq \psi_i(\mathcal{T})$, we have $\gamma_j^{\psi_i(\mathcal{T}')+1} \leq \gamma_j^{\psi_i(\mathcal{T})+1}$, and thus, $\Delta(\mathcal{T}, v) \geq \Delta(\mathcal{T}', v)$. On the other hand if $v \notin W_i$ then $\psi_i(\mathcal{T}' \cup v) - \psi_i(\mathcal{T}') = \psi_i(\mathcal{T} \cup v) - \psi_i(\mathcal{T}) = 0$ and $\Delta(\mathcal{T}, v) = \Delta(\mathcal{T}', v) = 0$, i.e., adding v does not change the vertex count ψ_i and thus the value of the basis function $r_{i,j}$. Since this holds for all W_i, we obtain $R(\mathcal{T}' \cup v) - R(\mathcal{T}') \leq R(\mathcal{T} \cup v) - R(\mathcal{T})$ and R is submodular. $\square$

Algorithm 1: Bi-criterion approximation for GTOP

   Input: $G = (V, E, l)$, start $s \in V$, #robots m, budget
        B, weights $\boldsymbol{w}$
   Output: Tours $\mathcal{T}$
1 Initialize $\boldsymbol{\psi} = \boldsymbol{0}$, $\mathcal{T} \leftarrow \emptyset$
2 **for** $k = 1$ *to* m **do**
3 | $T_k \leftarrow \textsc{SingleOP}(G, s, \boldsymbol{w}, \boldsymbol{\psi})$
4 | **for** $i = 1$ *to* n **do**
5 | | $\psi_i \leftarrow \psi_i + |W_i \cap V(T_k)|$
6 | $\mathcal{T} \leftarrow \mathcal{T} \cup \{T_k\}$
7 **return** $\mathcal{T}$

B. Solving the GTOP for a Given Set of Weights

In [5], authors provide a bi-criterion approximation algorithm for the orienteering problem with submodular rewards, and in [20] propose an approximation algorithm to extend the results of the orienteering problem to the team orienteering problem. In the rest of the paper, for a given $\boldsymbol{w}$, we combine these two approaches to achieve a bi-criterion approximation algorithm for the team orienteering problem with submodular reward functions. Algorithm 1 shows the proposed approach for the GTOP problem. The algorithm sequentially solves the submodular orienteering problem with the algorithm proposed in [5] (line 3). Then after iteration k, ψ_i is incremented by the number of vertices T_k visits in W_i (line 5). At each iteration, the SINGLEOP implements the proposed bi-criterion approximation algorithm in [5] with approximation factor $\eta = 2(1 - \frac{1}{e})^{-1}$ on the collected rewards. Hence, by Theorem 1 in [20], Algorithm 1 is a $1 + \eta$ approximation algorithm for the GTOP problem.

IV. LEARNING REWARDS FROM CHOICE FEEDBACK

One framework for learning reward functions via user interaction that found widespread attention in HRI in recent years is *learning from choice*. Iteratively, the robots present the user with two alternative solutions to some robot planning problems. The user then chooses the preferred option. The user is assumed to make that choice based on some hidden reward function which allows the robot to infer the parameters of that reward function.

A. Deterministic user feedback

We begin by posing our problem for a deterministic user, whose feedback always follows the assumed cost function. Consider that two sets of tours $\mathcal{T}^1$ and $\mathcal{T}^2$ are proposed to the user, and the user indicates their preference.

Definition 1 (Deterministic user model)**.** Given two sets of tours $\mathcal{T}^1$ and $\mathcal{T}^2$, a deterministic user always prefers the tours with larger reward with respect to the hidden user weights $\boldsymbol{w}^*$. Let $I \in \{1, 2\}$ denote the user feedback. Then

$$R(\mathcal{T}^1, \boldsymbol{w}^*) \geq R(\mathcal{T}^2, \boldsymbol{w}^*) \iff I = 1. \quad (5)$$

Learning Cuts: Without loss of generality, assume that the user prefers $\mathcal{T}^1$ (we can simply reassign the labels after

observing the choice), therefore we have, $\phi(\mathcal{T}^1) \cdot \boldsymbol{w} - \phi(\mathcal{T}^2) \cdot \boldsymbol{w} \geq 0$. We refer to this inequality as a *cut*. Let $P(c_{1:k})$ denote the polyhedron constructed by the cuts $\{c_1, \ldots, c_k\}$. Now we define a *valid cut* as follows:

Definition 2 (Valid Cut)**.** Given a polyhedron $P(c_{1:k})$, a cut c_{k+1} is valid if the intersection of the cut and $P(c_{1:k})$ has dimension greater than zero.

Note that each valid cut partitions the space of valid rewards $\boldsymbol{w}$, therefore by adding a valid cut at each step we monotonically decrease the set of valid rewards. Now assume that we have a tours $\mathcal{T}^1$ in hand, we want to construct the set $\mathcal{T}^2$ such that the tours in $\mathcal{T}^2$ satisfy the budget constraints and the cut constructed by comparing $\mathcal{T}^1$ and $\mathcal{T}^2$ is valid.

Lemma 1. Given two set of tours $\mathcal{T}^1$ and $\mathcal{T}^2$ and a set of prior cuts $\{c_1, \ldots, c_k\}$, the cut constructed by comparing $\mathcal{T}^1$ and $\mathcal{T}^2$ is valid if and only if the solutions to the following problems are greater than zero:

$$\max_{\boldsymbol{w}} \phi(\mathcal{T}^1)\boldsymbol{w} - \phi(\mathcal{T}^2)\boldsymbol{w} \quad \text{subject to: } \boldsymbol{w} \in P(c_{1:k}),$$

$$\max_{\boldsymbol{w}} \phi(\mathcal{T}^2)\boldsymbol{w} - \phi(\mathcal{T}^1)\boldsymbol{w} \quad \text{subject to: } \boldsymbol{w} \in P(c_{1:k}).$$

Proof. The first part is trivial. Now assume that the cut is valid, then the cut intersects the interior of $P(c_{1:k})$. Let $\boldsymbol{d}$ be the vector normal to the line defined by the cut, then there exists a $\boldsymbol{w}_0$, δ_1 and δ_2 such that $\boldsymbol{w}_0 + \delta_1 \boldsymbol{d} \in P(c_{1:k})$ and $\boldsymbol{w}_0 - \delta_2 \boldsymbol{d} \in P(c_{1:k})$. Therefore, the solution to the two problems are greater than zero. $\qquad\square$

In essence, Lemma 1 states that for a valid cut $(\mathcal{T}^1, \mathcal{T}^2)$ there must exist some $\boldsymbol{w}$ in the current polyhedron $P(c_{1:k})$ for which $\mathcal{T}^1$ has a higher reward than $\mathcal{T}^2$, and vice versa. In other words, the hyperplane defining a valid cut passes through the interior of the current polyhedron $P(c_{1:k})$.

Query generation: The main challenge in active preference learning is to iteratively generate valid cuts that allow for efficient learning.

Related work in HRI is usually based on heuristic solutions that greedily optimize some auxiliary function h to maximize the expected learning benefit of presenting two solutions $(\mathcal{T}^1, \mathcal{T}^2)$. Recent approaches include h capturing the volume of the probability space over weights [7], the information entropy [9] or the maximum regret [11].

These optimizations are usually difficult on two different levels: Computing h often poses a hard problem and the potential solutions require solving some robot planning problem, making this a nested optimization. Most solution techniques rely on sampling candidates solutions, as well as approximating h using sampled weights. While this might be suitable for low-dimensional applications, the number of basis functions for team orienteering problems of our problem is $O(n|\Gamma|)$. Thus, accurately approximating information entropy requires a prohibitively large number of samples.

We design a novel query generation method that does not require any form of sampling. Similar to [10] we choose a variation of learning from choice in which one of the two presented options comes from the previous iteration: At

iteration k, let $\mathcal{T}^{\text{curr}}$ be the tours the user preferred in the previous iteration. We now need to find only one new set of tours $\mathcal{T}^{\text{new}}$ such that observing feedback to $(\mathcal{T}^{\text{curr}}, \mathcal{T}^{\text{new}})$ yields a valid cut with respect to $\{c_1, \ldots, c_k\}$.

To find $\mathcal{T}^{\text{new}}$ given the previous cuts $\{c_1, \ldots, c_k\}$ and $\mathcal{T}^{\text{curr}}$, we adapt the maximum regret approach proposed in [11]. Regret measures how suboptimal the solution of estimated parameters w' is. In the GTOP, this is captured by the reward of some tours $\mathcal{T}'$, evaluated by the users true reward function w^* compared against the user-optimal solution $\mathcal{T}^*$, evaluated by w^*, i.e., $R(\mathcal{T}^*, w^*) - R(\mathcal{T}', w^*) = \phi^* w^* - \phi' w^*$. Using regret we can find $\mathcal{T}^{\text{new}}$ by solving

$$\max_{w^{\text{new}}} \phi(\mathcal{T}(w^{\text{new}})) w^{\text{new}} - \phi(\mathcal{T}^{\text{curr}}) w^{\text{new}}$$
$$\text{subject to: } w^{\text{new}} \in P(c_{1:k-1}). \quad (6)$$

That is, given the current solution, we seek to find $\mathcal{T}^{\text{new}}$ such that if $\mathcal{T}^{\text{new}}$ was optimal, the current solution would be most suboptimal, i.e., have maximum regret. If the user chooses $\mathcal{T}^{\text{curr}}$, the weight w^{new} becomes infeasible thus $(w^{\text{curr}}, w^{\text{new}})$ will no longer be the maximizer for the updated polyhedron $P(c_{1:k})$ – we greedily reduce the upper bound on the error. On the other hand, if the user chooses $\mathcal{T}^{\text{new}}$, we improve the current solution. This formulation makes two major changes to the max regret approach in [11]: 1) we fix one set of tours to be shown to be $\mathcal{T}^{\text{curr}}$, and 2) we use the difference instead of a ratio in the definition of regret. The following proposition ensures that an algorithm that iteratively solves (6) and then updates the polyhedron given the user feedback will eventually find an optimal solution, i.e., a weight w^{curr} where $R(\mathcal{T}(w^{\text{curr}})) = R(\mathcal{T}(w^*))$.

Proposition 1. If the optimal solution to Problem (6) is not a valid cut, then the reward collected by $\mathcal{T}^{\text{curr}}$ is optimal.

Proof. Let w^{new} be the optimal solution to Problem (6). Since the cut defined by $\mathcal{T}^{\text{curr}}$ and $\mathcal{T}(w^{\text{new}})$ is not a valid cut, then we have $\phi(\mathcal{T}(w^*)) w^* - \phi(\mathcal{T}^{\text{curr}}) w^* \leq \phi(\mathcal{T}(w^{\text{new}})) w^{\text{new}} - \phi(\mathcal{T}^{\text{curr}}) w^{\text{new}} = 0$, where the first inequality comes from $w^* \in P(c_{1:k-1})$ and the second equality comes from Lemma 1. Therefore, $\mathcal{T}^{\text{curr}}$ collects the optimal reward. $\square$

Now we establish the following result on the complexity of Problem (6).

Lemma 2. The problem of finding the tour with maximum regret is NP-hard.

Proof. We show the result by a reduction from the traveling salesman problem (TSP). Given a TSP instance $G = (V, E, c)$ and a budget B, we construct an instance of problem (6). We set the polyhedron P to be unit cube and T^{curr} to be the set of empty tours. Then problem (6) becomes $\max_{w^{\text{new}}} \phi(\mathcal{T}(w^{\text{new}})) w^{\text{new}}$ and budget B on the tours. Let w^* be the optimal solution to this problem, then $\mathcal{T}(w^*)$ where $\phi(\mathcal{T}(w^*)) w^* = |V|$ is a valid solution to the TSP. Now note that if there is no solution to the max regret problem collecting $|V|$ reward, then there is no solution to the TSP problem, therefore, the result follows immediately. $\square$

We observe the following property of the objective function in (6) which will help us provide a bound on it.

Lemma 3. The objective function $\phi(\mathcal{T}(w)) w - \phi(\mathcal{T}') w$ is a convex function in w for any set of tours $\mathcal{T}'$.

Proof. Consider $w = \lambda w^1 + (1 - \lambda) w^2$ for some $\lambda \in [0, 1]$. Then,

$$\phi(\mathcal{T}) w = \phi(\mathcal{T}(w))[\lambda w^1 + (1 - \lambda) w^2]$$
$$= \lambda \phi(\mathcal{T}(w)) w^1 + (1 - \lambda) \phi(\mathcal{T}(w)) w^2$$
$$\leq \lambda \phi(\mathcal{T}(w^1)) w^1 + (1 - \lambda) \phi(\mathcal{T}(w^2)) w^2.$$

Note that the second term in the objective function is linear in w. Therefore, the result follows immediately. $\square$

While Lemma 2 shows that finding the set of tours maximizing the regret is NP-hard, Lemma 3 implies that the optimal solution of (6) is on a vertex of the polyhedron $P(c_{1:k})$. We can upper bound that solution with

$$\min_{w^{\text{new}}} \phi(\mathcal{T}^{\text{curr}}) \cdot w^{\text{new}} \quad \text{subject to: } w^{\text{new}} \in P(c_{1:k}). \quad (7)$$

In conclusion, at iteration k our min-max regret heuristic proposes two new sets of tours $(\mathcal{T}^{\text{curr}}, \mathcal{T}^{\text{new}})$ where $\mathcal{T}^{\text{curr}}$ is the solution the user preferred in the previous iteration, and $\mathcal{T}^{\text{new}}$ is the approximate GTOP solution for w^{new} solving (7).

B. Extension to noisy user feedback

In the previous section, we considered the problem with a deterministic user who always chooses the set of tours with higher reward with respect to w^*. In practice, this assumption can lead to suboptimal outcomes when the user decision is not accurately captured in the assumed reward function. Thus, we consider the problem with a noisy user where the set of tours chosen by the user is not the set of tours collecting higher rewards. We model the noisy user with the Boltzmann model as follows:

Definition 3 (Noisy user model). Given two sets of tours $\mathcal{T}^1$ and $\mathcal{T}^2$, and a user with hidden rewards w, then the probability that the user chooses $\mathcal{T}^1$ is

$$\mathbb{P}(\mathcal{T}^1, w) = \frac{1}{1 + \exp(\beta(\phi(\mathcal{T}^2) - \phi(\mathcal{T}^1)) \cdot w)},$$

where $\beta > 0$ represents the level of expertise of the user.

The Boltzmann model is widely used in reward learning [7], [9], [13], [32] and describes a user whose choice becomes more uncertain when the presented options have a similar reward with respect to w^*.

Now consider a set of cuts $\{c_1, \ldots, c_k\}$ which are results of the preference questions, then the probability that the hidden reward function lies in $P(c_{1:k})$ is $\mathbb{P}(w^* \in P(c_{1:k})) = \Pi_{i=1}^k \mathbb{P}(c_i)$, where $\mathbb{P}(c_i)$ is the probability that the user has responded to the ith query correctly. We denote the negation of a cut c_i by $\bar{c}_i$ and the probability of it as $\mathbb{P}(\bar{c}_i) = 1 - \mathbb{P}(c_i)$.

Algorithm 2 shows the proposed algorithm for learning the reward function of the user. In Line 1 of the algorithm, we initialize the set of observed cuts to $Q = \emptyset$.

In Line 4, function PROBABLEREGIONS(Q, N) takes the current set of cuts $Q = \{c_1, \ldots, c_i\}$ and an integer N as input and returns a set of N polyhedrons. Each such polyhedron is constructed as follows: We initialize a set of cuts $Q' =$

Algorithm 2: Learning GTOP Rewards

Input: graph $G = (V, E, l)$, start $s \in V$, fleet size m,
     budget B, sample budget N

Output: Tours $\mathcal{T}$

1 Initialize $\boldsymbol{w} = \boldsymbol{1}$, $Q \leftarrow \emptyset$
2 $\mathcal{T} = \text{GTOP}(G, \boldsymbol{w})$
3 for $i = k$ *to* K **do**
4 $\{P_1, \ldots, P_N\} \leftarrow \text{PROBABLEREGIONS}(Q, N)$
5   **for** $P_j \in \{P_1, \ldots, P_N\}$ **do**
6 $\boldsymbol{w}^j \leftarrow \text{MAXIMUMREGRET}(G, \mathcal{T}, P^j)$
7 $\mathcal{T}^j \leftarrow \text{GTOP}(G, \boldsymbol{w}^j)$
8 $\mathcal{T}_{\text{new}} \leftarrow \arg\max_{\mathcal{T}^j} \mathbb{P}(\boldsymbol{w}^* \in P_j)(\boldsymbol{\phi}(\mathcal{T}^j) - \boldsymbol{\phi}(\mathcal{T})) \cdot \boldsymbol{w}^j$
9 $\mathcal{T}, c_k \leftarrow \text{USERRESPONSE}(\mathcal{T}, \mathcal{T}^{\text{new}})$
10 $Q \leftarrow Q \cup \{c_k\}$
11 return $\mathcal{T}$

Q. Then for each cut $c_i \in Q'$, we replace c_i with $\bar{c}_i$ with probability $1 - \mathbb{P}(c_i)$. This then defines a new polyhedron $P(Q')$. That is, we sample from the set of all 2^K possible combinations of cuts or their negations. Thus, the probability of sampling a set of cuts Q' and thus a polyhedron $P(Q')$ is $\mathbb{P}(\boldsymbol{w}^* \in P(Q'))$. Considering multiple polyhedrons allows us to take into account inconsistency in the answers by the user.

For each of the constructed polyhedrons, function MAXIMUMREGRET solves Problem (7). We generate a set of tours for each of the rewards as a candidate sets of tours for the next preference questions. Finally, in Line 8 we evaluate the maximum regret for each polyhedron and discount it by their probabilities. The level of expertise for the user and the reward function are not known, however observe that by Definition 3 we have $\mathbb{P}(c_i) > 1/2$. Therefore, we approximate the probabilities of regions as a monotonically decreasing function of the number of negated cuts in the construction of the polyhedron. Finally, the sets of tours with the highest discounted regret is presented to the user as a new query.

V. EVALUATION

We evaluate the performance in environmental monitoring missions using real-world and randomly generated scenarios.

A. Experiment Setup

In the experiments, the robot fleet is given a set of regions of interest, but no information on how valuable it is to collect environmental data in each region. The objective is to learn a reward function describing which regions are to be prioritized when battery life does not allow to visit all of them.

Basis functions: We define basis functions that capture the visits to each of the regions of interest. In each region, we randomly place 1 to 5 vertices. For each set W_i of vertices in a single region we define three different basis functions as in (3) for decay parameters $\gamma \in \{.001, .5, 1\}$. That is, the user reward for each region can follow a step function, a curved submodular function, or a linear function.

User design: Drawing user weights for the basis functions uniformly random does not yield relevant problem instances: the initial solution $\mathcal{T} = \text{GTOP}(\text{G}, \boldsymbol{w} = \boldsymbol{1})$ is often already close to the optimal solution.

Thus, we design a probability function for how a user places weights on these basis functions. First, we model the user weight w_i as a function of the distance of the region V_i from the depot. In detail, the weight is a Gaussian random variable $w_i = \mathcal{N}(d(s, V_i)^2, \sigma)$ where $d(s, V_i)$ is the distance from the start vertex to the mean location of V_i, and the variance σ is a design parameter A second parameter $\boldsymbol{p}$ describes a probability over the different values of γ—for each region the user "picks" only one of the three decay parameters. Thus, the vector $\boldsymbol{p} = \begin{bmatrix} p_1, p_2, p_3 \end{bmatrix}$ where $p_1 + p_2 + p_3 = 1$ describes the probabilities for γ taking values $\{.001, .5, 1\}$, respectively.

We simulate users with values $\sigma \in \{.5, 10\}$ representing scenarios with moderate and almost no correlation of reward and distance, and distributions over decay functions $\boldsymbol{p} \in \{[1/3, 1/3, 1/3], [.7, .2, .1], [.1, .2, .7], [.2, .7, .1]\}$ where the weights represent a bias towards step, linear and submodular functions, respectively. Thus, our simulated users vary in the structure of what regions are important to them, as well as in the type of observations they are interested in, i.e., repeated observations of the same region, or rather covering more regions with fewer or even just one observation. For all 8 different parameter settings, we choose $\beta = 20$. This results in users choosing the set of tours with higher rewards in 84% of uniformly random queries.

We consider all robots start at the same depot and have a budget equal to twice the distance from the depot to the furthest region. In the implementation of Algorithm 2, we use a simple static probability $\mathbb{P}(c_i) = 0.8$ and $N = 10$ PROBABLEREGIONS. We measure how well Algorithm 2 learns the weights $\boldsymbol{w}^*$ of the user reward function, i.e., solves Problem 3. Let $\mathcal{T}$ be the set of tours returned by Algorithm 2, the reward ratio is $R(\mathcal{T}, \boldsymbol{w}^*)/R(\mathcal{T}^*, \boldsymbol{w}^*)$, where $\boldsymbol{w}^*$ are the hidden user weights, and $\mathcal{T}^*$ is the corresponding approximate GTOP solution using Algorithm 1. We notice that $\mathcal{T}^*$ is only an approximation to the optimum; thus, the ratio can be larger than 1.

B. Baselines

We compare the proposed maximum regret heuristic against two classes of baselines.

Richer user input: The first class of baselines consists of non-learning approaches. Instead, we consider that the user provides more information about their reward function to the robot. In the proposed framework, the user only identifies regions of interest such as the protected areas in Figure 1. Without any learning, the robot fleet assumes equal reward for all regions, and that the decay can be either linear, submodular, or a step function. This constitutes the initial solution of our algorithm. With increasing complexity of user input we consider: Decay – users do not provide numerical reward values for the regions, but indicate what decay function each region has, $\text{Ranking} + \text{Decay}$ – the user gives a ranking of the importance of each region, and the decay function, Reward –

Fig. 2: Example tours from a common depot (blue dot) for the environmental monitoring task. The left figure shows the initial solution before learning with $\boldsymbol{w} = \boldsymbol{1}$. The middle figure shows the tours that were learned after 20 iterations, while the right figure shows the approximately optimal tours $\mathcal{T}^*$.

the user provides the exact rewards for the region, but not the decay, and finally `Reward + Decay` where the user provides the rewards and the decay, which is equivalent to providing $\boldsymbol{w}^*$. The latter two methods are effectively reward designs, which require a very high level of expertise from the user and thus are often impractical.

Competing active learning methods: The second class of baselines consists of competing approaches for the active query generation. The first is `Random Uniform` where we replace line 9 of Algorithm 2 with computing $\mathcal{T}^{\text{new}}$ for some uniformly randomly sampled weight $\boldsymbol{w}^{\text{new}}$. A second method `Random Posterior` samples based on previously observed user feedback. We replace the direction $\phi(\mathcal{T}^{\text{curr}})$ of the objective function in (7) with some uniformly randomly sampled direction $\boldsymbol{d} \in [-1, 1]^n$, and set $N = 1$ in line 5 of Algorithm 2, i.e., generate only one probable region P.

The third query generation method `Information Gain` adapts the information entropy approach proposed in [9]. However, the original algorithm is unsuccessful in our problem: It returns the expected weight, which is often rendered 0 as the weight samples do insufficiently cover the high dimensional space. Thus, we adapted the entropy approach to also follow our framework where we show the previously preferred option $\mathcal{T}^{\text{curr}}$ again. To find the second set of tours, we compute a set of candidates $\mathcal{T}_1, \ldots, \mathcal{T}_k$: We execute lines 5 to 8 of our Algorithm 2, but replace line 7 with a linear program optimizing in a random direction, similar to `Random Posterior`. We then select the best candidate by solving Equation (4) from [9], using $M = 300$ weight samples. Each sample is drawn uniformly random from a sampled polyhedron returned by PROBABLEREGIONS$(Q, 1)$.

C. Results

Real world environment with two robots: In the first experiment, we use part of the World Database on Protected Areas (WDPA)[1], illustrated in Figure 2. The problem contains 17 regions, and for each of the 8 user types we simulated 25 trials.

[1]Dataset from https://data.unep-wcmc.org/datasets/12

(a) Comparison with richer user input

(b) Comparison with different learning methods.

Fig. 3: Summarized results for the real world experiment comparing the proposed approach against levels of a-priori user input (a), and against other active preference learning methods (b).

Figure 3 shows the comparison of our proposed method against the two baseline classes. In Figure 3a we observe that `Maximum Regret` drastically improves the reward ratio within the first few iterations, finding tours as good as `Ranking + Decay`, where the user would provide a noiseless ranking of all regions and identify the decay function. After 12 iterations the learning collects as much reward as `Reward` and approaches `Reward + Decay` (i.e., the ground truth) after 20 iterations. This showcases that the learning approach is able to find tours equally good to those that require much more complex user input when not using preference learning. Furthermore, after 20 iterations the learned set of tours collects 95% of the reward of the approximately optimal tours. Figure

3b shows that `Maximum Regret` collects the most reward after 20 iterations (95% compared to 82% for `Information Gain`, 75% for `Random Posterior` and 68% for `Random Uniform`). Moreover, there is a strong difference in the learning speed: After only 2 iterations `Maximum Regret` already achieves 82%, matching the result for `Information Gain` after 18 iterations.

Synthetic environment with 4 robots: To assert that the real-world experiment is representative of a wider range of problem instances, we consider a synthetic experiment where the regions of interest are randomly generated. We construct additional problem instances by sampling 20 regions of interest, each offering 1 to 5 observations points. We observed similar performance on the synthetic instances: After 10 iterations the proposed method collects as much reward as `Ranking + Decay` and `Reward` (81%). After 20 iterations our method achieves a reward ratio of 89% compared to 72% for `Random Posterior`, 70% for `Information Gain`, and 68% for `Random Uniform`.

In summary, the proposed method finds high-quality tours for the data collection task that, when no learning was used, would require much richer user input; and our method outperforms other learning approaches in terms of collected reward and learning speed.

VI. DISCUSSION

This paper considers the problem of reward collection by a team of robots with a hidden submodular reward function. First, we presented a linear approximation of the submodular reward function. Second, we proposed an approximation algorithm when the weights on the linear approximation is known. Finally, we proposed a framework to learn the underlying hidden user reward function for deterministic and noisy users from choice feedback. The experimental results on real-world data show that the proposed framework provides near-optimal tours after a few iterations of user feedback. For future work, we consider investigating the effectiveness of the proposed method in data collection missions in the field.

REFERENCES

[1] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, "The orienteering problem: A survey," *European Journal of Operational Research*, vol. 209, no. 1, pp. 1–10, 2011.

[2] B. Liu, X. Xiao, and P. Stone, "Team orienteering coverage planning with uncertain reward," *arXiv preprint arXiv:2105.03721*, 2021.

[3] S. Jorgensen, R. H. Chen, M. B. Milam, and M. Pavone, "The team surviving orienteers problem: routing teams of robots in uncertain environments with survival constraints," *Autonomous Robots*, vol. 42, no. 4, pp. 927–952, 2018.

[4] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and its application to informative path planning for persistent monitoring tasks," in *2014 IEEE/RSJ IROS*. IEEE, 2014, pp. 342–349.

[5] H. Zhang and Y. Vorobeychik, "Submodular optimization with routing constraints." in *AAAI*, vol. 16, 2016, pp. 819–826.

[6] H. J. Jeon, S. Milli, and A. D. Dragan, "Reward-rational (implicit) choice: A unifying formalism for reward learning," *arXiv preprint arXiv:2002.04833*, 2020.

[7] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia, "Active preference-based learning of reward functions," in *RSS*, 2017.

[8] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 1.

[9] B. Erdem, M. Palan, N. C. Landolfi, D. P. Losey, and D. Sadigh, "Asking easy questions: A user-friendly approach to active reward learning," PMLR, pp. 1177–1190, 2020.

[10] N. Wilde, A. Blidaru, S. L. Smith, and D. Kulić, "Improving user specifications for robot behavior through active preference learning: Framework and evaluation," *IJRR*, vol. 39, no. 6, pp. 651–667, 2020.

[11] N. Wilde, D. Kulić, and S. L. Smith, "Active preference learning using maximum regret," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 10 952–10 959.

[12] Y. Cui and S. Niekum, "Active reward learning from critiques," in *2018 IEEE International Conference on Robotics and Automation*, May 2018, pp. 6907–6914.

[13] D. Brown, R. Coleman, R. Srinivasan, and S. Niekum, "Safe imitation learning via fast bayesian reward inference from preferences," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1165–1177.

[14] M. Korein and M. Veloso, "Multi-armed bandit algorithms for spare time planning of a mobile service robot," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2018, pp. 2195–2197.

[15] A. Jain, S. Sharma, T. Joachims, and A. Saxena, "Learning preferences for manipulation tasks from online coactive feedback," *IJRR*, vol. 34, no. 10, pp. 1296–1313, 2015.

[16] A. Wilson, A. Fern, and P. Tadepalli, "A bayesian approach for policy learning from trajectory preference queries," *Advances in neural information processing systems*, vol. 25, pp. 1133–1141, 2012.

[17] A. Shah, S. Wadhwania, and J. Shah, "Interactive robot training for non-markov tasks," *arXiv preprint arXiv:2003.02232*, 2020.

[18] R. Holladay, S. Javdani, A. Dragan, and S. Srinivasa, "Active comparison based learning incorporating user uncertainty and noise," in *RSS Workshop on Model Learning for Human-Robot Communication*, 2016.

[19] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—i," *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.

[20] A. Singh, A. Krause, C. Guestrin, W. J. Kaiser, and M. A. Batalin, "Efficient planning of informative paths for multiple robots," in *IJCAI*, vol. 7, 2007, pp. 2204–2211.

[21] I. Dolinskaya, Z. E. Shi, and K. Smilowitz, "Adaptive orienteering problem with stochastic travel times," *Transportation Research Part E: Logistics and Transportation Review*, vol. 109, pp. 1–19, 2018.

[22] A. M. Campbell, M. Gendreau, and B. W. Thomas, "The orienteering problem with stochastic travel and service times," *Annals of Operations Research*, vol. 186, no. 1, pp. 61–81, 2011.

[23] L. Bottarelli, M. Bicego, J. Blum, and A. Farinelli, "Orienteering-based informative path planning for environmental monitoring," *Engineering Applications of Artificial Intelligence*, vol. 77, pp. 46–58, 2019.

[24] R. N. Smith, M. Schwager, S. L. Smith, B. H. Jones, D. Rus, and G. S. Sukhatme, "Persistent ocean monitoring with underwater gliders: Adapting sampling resolution," *Journal of Field Robotics*, vol. 28, no. 5, pp. 714–741, 2011.

[25] A. B. Asghar, S. L. Smith, and S. Sundaram, "Multi-robot routing for persistent monitoring with latency constraints," in *IEEE American Control Conference*, 2019, pp. 2620–2625.

[26] A. Sadeghi, A. B. Asghar, and S. L. Smith, "On minimum time multi-robot planning with guarantees on the total collected reward," in *IEEE International Symposium on Multi-Robot and Multi-Agent Systems*, 2019, pp. 16–22.

[27] T. Somers and G. A. Hollinger, "Human–robot planning and learning for marine data collection," *Autonomous Robots*, vol. 40, no. 7, pp. 1123–1137, 2016.

[28] T. Somers, N. R. Lawrance, and G. A. Hollinger, "Efficient learning of trajectory preferences using combined ratings and rankings," in *Proc. RSS Workshop on Mathematical Models, Algorithms, and Human-Robot Interaction , Boston, MA*, 2017.

[29] W. Xu, W. Liang, Z. Xu, J. Peng, D. Peng, T. Liu, X. Jia, and S. K. Das, "Approximation algorithms for the generalized team orienteering problem and its applications," *IEEE/ACM Transactions on Networking*, 2020.

[30] J. Das, F. Py, J. B. Harvey, J. P. Ryan, A. Gellene, R. Graham, D. A. Caron, K. Rajan, and G. S. Sukhatme, "Data-driven robotic sampling for marine ecosystem monitoring," *The International Journal of Robotics Research*, vol. 34, no. 12, pp. 1435–1452, 2015.

[31] A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, and C. Faloutsos, "Efficient sensor placement optimization for securing large water distribution networks," *Journal of Water Resources Planning and Management*, vol. 134, no. 6, pp. 516–526, 2008.

[32] C. Basu, M. Singhal, and A. D. Dragan, "Learning from richer human guidance: Augmenting comparison-based learning with feature queries," in *International Conference on Human-Robot Interaction*. NY, USA: ACM, 2018, pp. 132–140.