

Online Pareto-Optimal Decision-Making for Complex Tasks using Active Inference

Peter Amorese[†], Shohei Wakayama[†], Nisar Ahmed, and Morteza Lahijanian

Abstract—When a robot autonomously performs a complex task, it frequently must balance competing objectives while maintaining safety. This becomes more difficult in uncertain environments with stochastic outcomes. Enhancing transparency in the robot’s behavior and aligning with user preferences are also crucial. This paper introduces a novel framework for multi-objective reinforcement learning that ensures safe task execution, optimizes trade-offs between objectives, and adheres to user preferences. The framework has two main layers: a multi-objective task planner and a high-level selector. The planning layer generates a set of optimal trade-off plans that guarantee satisfaction of a temporal logic task. The selector uses active inference to decide which generated plan best complies with user preferences and aids learning. Operating iteratively, the framework updates a parameterized learning model based on collected data. Case studies and benchmarks on both manipulation and mobile robots show that our framework outperforms other methods and (i) learns multiple optimal trade-offs, (ii) adheres to a user preference, and (iii) allows the user to adjust the balance between (i) and (ii).

Index Terms—Multi-Objective Decision Making, Active Inference, Formal Synthesis

I. INTRODUCTION

THE demand for robots to autonomously perform hazardous and repetitive tasks is steadily increasing [1, 2]. Such tasks often involve multiple, possibly competing, quantitative objectives, e.g., time and energy, requiring optimization for trade-offs between the objectives, known as *Pareto optimality*. In unknown environments, like the Martian surface or civilian households, however, such quantities are often stochastic and unknown *a priori*. Fortunately, they are measurable and can be learned online. The robot should make decisions to learn *multiple* optimal trade-offs, as well as focus on learning a single *user-preferred* trade-off. Although highly desirable, accomplishing both simultaneously is challenging since the trade-off selection must balance exploring unknown trade-offs with exploiting preferred trade-offs. This paper focuses on this challenge and develops a framework that simultaneously (i) learns multiple optimal trade-offs, (ii) embeds an intuitive user preference over a desired trade-off, and (iii) allows the user to adjust the balance between (i) and (ii).

Example 1. Consider the robotic dishwashing scenario in Figure 1. The robotic manipulator is equipped with a sampling-based motion planner (e.g., RRT and PRM [3, 4]) that solves A-to-B planning queries. The robot must repetitively complete a complex task of inserting the dishes into

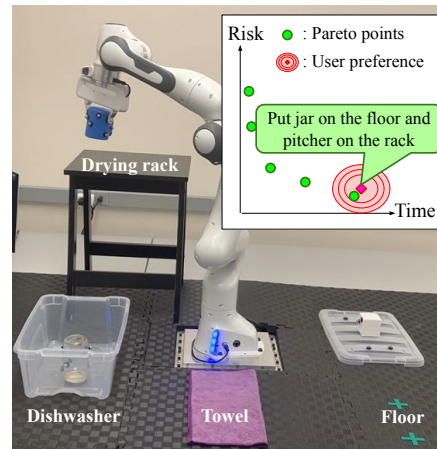


Fig. 1: Motivating robotic dishwashing scenario: a robotic manipulator needs to compare execution time of a task and inherent risks of dropping fragile dishes (the jar is more fragile than the blue pitcher). The user’s preferred trade-off between time and risk is incorporated for transparent behaviors.

the dishwasher, then properly drying them by either placing directly up on the rack, or placing on the towel and moving them to the floor. In such a scenario, the robot should be both efficient (minimize the time of execution) and risk-aware when carrying fragile dishes high above the ground. However, due to the nature of sampling-based methods, the characteristics of actions (time and path-height) are unknown *a priori*, and hence the robot needs to construct accurate models of these quantities during deployment. The robot can either quickly place dishes up on a drying rack (low-time, high-risk for fragile objects), or manually dry dishes on the ground (high-time, low-risk). Considering a human user may prefer one trade-off over the other, the robot must both learn the quantitative models, and decide amongst optimal trade-off actions while adhering to the user’s preference. In this work we generalize this scenario using an information-theoretic approach to reasoning over such cost functions that are unknown, multivariate, and stochastic.

Embedding the satisfaction of a task into the (multi-objective) reward function often aims to capture both qualitative properties of the task (i.e., whether or not the robot completed the task), as well as quantitative properties (i.e., how well the robot completed the task) [5]. However, properly representing qualitative task completion through a reward function is a challenging problem, often requiring the use of methods such as inverse reinforcement learning (IRL) [6]. In uncertain environments, it is not always possible to obtain large amounts of human teaching data. Furthermore, these

[†] Authors contributed equally to this work

Authors are with the University of Colorado Boulder.

learning-based methods often fail to provide safety guarantees on the qualitative behavior of the robot. Nevertheless, in scenarios where qualitative aspects of the robotic model and task are known beforehand, e.g., manipulation problems, *formal methods* [7] provide alternative approaches to both task representation and plan synthesis using temporal logic specifications, such as Linear Temporal Logic (LTL) [8] and LTL over finite traces (LTLf) [9]. These planning methods can solve the qualitative portion of the learning problem and provide guarantees on the completion of tasks and safety while avoiding the need for IRL. Formal synthesis approaches have also been used to solve multi-objective quantitative planning problems [10]; however, those approaches assume a perfectly accurate quantitative model, which is often unavailable.

Decision making problems involving competing objectives can be formulated as multi-objective reinforcement learning (MORL) [11, 12, 13, 14]. Unlike single-objective RL, MORL agents must select among potentially many optimal trade-offs. The numerical valuations of these optimal trade-offs are referred to as the Pareto front. Learning all optimal trade-offs yields valuable insight into the trade-off analysis of the learned results. However, in some situations, only local *preferred* subregions of the Pareto front may be of interest [15], allowing the learner to avoid spending effort learning unpreferred trade-offs. The Multi-Objective Multi-Armed Bandit (MOMAB) community studies means of learning all optimal trade-off actions [16, 17, 18, 19]. Alternatively, one can use *scalarization* to convert the multi-objective problem into a single objective problem based on a *preference* trade-offs [11, 12, 13, 14]. Further description of these methods can be found in Sec. V. To the best of our knowledge, no MORL selection methods simultaneously propose mathematically sound adherence to a *preferred* trade-off while learning and exploring portions of the true Pareto front.

In light of these gaps, this study proposes a novel MORL framework centered around viewing optimal trade-off selection as its own high-level decision making under uncertainty problem. Specifically, we employ *active inference* (AIF), recently explored as a sound approach for sequential decision-making under uncertainty [20, 21, 22, 23] that minimizes an information-theoretic quantity known as *surprise*. Loosely speaking, surprise quantifies how well a probabilistic model can describe an observed outcome. By viewing multi-objective utility associated with completing the task (in our case a cost vector) as the probabilistic outcome, we exploit the decision making power of AIF to adhere to a user’s ideal trade-off while exploring and learning localized portions of the whole Pareto front. By encoding the aforementioned trade-off preference as a prior belief over outcomes, choosing plans that minimize surprise will not only learn the unknown costs, but also realize the user’s desired trade-off. To harness the rigorous benefits of surprise-minimization however, one must reason over the statistics of the transition cost model. In turn, the simple scenario in Example 1 may become very complex when little is known about the transition cost model *a priori*.

Directly optimizing surprise, however, introduces intractable computational complexity for finite-horizon tasks. Our framework integrates multi-objective *planning* for a complex LTLf

specification with high-level *selection* of an optimal trade-off by using a Bayesian approach to learning the quantitative model. We extend AIF to reasoning over finite horizon planning via upper-bounding the surprise with *expected free energy* (EFE). We derive computationally tractable techniques for optimizing EFE for LTLf tasks.

Using our framework, we illustrate the performance trade-offs between biasing towards a user’s preferred trade-off versus learning the entire Pareto front. Our information-theoretic approach to MORL lays the foundation for rigorously planning over a probabilistic cost-model while incorporating an intuitive formulation of the user’s preference. We showcase the utility of our approach using robotic case studies, including a hardware demonstration of the scenario in Example 1. Additionally, we show that our inference-based formulation can result in better sample efficiency for both achieving Pareto-optimal behavior as well as learning portions of the Pareto front.

The major contributions of this work are three-fold:

- A novel MORL framework that combines formally guaranteed task plan synthesis with efficient learning of Pareto-optimal behavior,
- Derivation of a tractable approximation of a *expected free energy* formulation for finite horizon planning over an uncertain quantitative model,
- Benchmarking the efficacy of the proposed AIF-based selection strategy against the state-of-the-art methods [12, 14, 19], along with Mars surface exploration simulation and a robotic hardware dish-washing experiment.

A detailed discussion on related work along with an overview of AIF are provided in Sec. V.

II. PRELIMINARIES AND PROBLEM FORMULATION

This paper studies a framework that enables a robot to (i) repetitively complete a given high-level complex task, and (ii) learn the most efficient and user-aligned means of completing the task. These two goals separate nicely into *qualitative* behavior, i.e., whether or not the robot completes the task, and *quantitative* behavior, i.e., how well the robot completes the task. We restrict our attention to problems where qualitative aspects of the model (i.e. robotic states, state transitions, task related observations, etc.) are known, but quantitative aspects (i.e. costs of executing robotic actions) are unknown and must be learned. Below, we first formally define a robot model and LTLf task specifications. We then connect these definitions with cost uncertainty and user preferences to be able to fully formulate the problem.

A. Robot Model

We consider a robotic system modeled as a deterministic transition system (DTS), an abstraction used in many formal approaches to robotic systems, including both mobile robots and robotic manipulators [7, 24, 25, 26, 27, 28, 29]. To capture uncertainty in quantitative values, we augment classical DTS with a stochastic cost function as defined below.

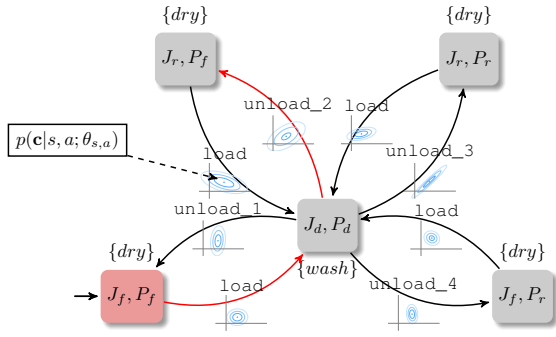


Fig. 2: The DTS-sc T described in Example 2 is shown. Each state-action transition is associated with an (example) unique cost distribution shown in blue. A plan that satisfies the task described in Example 3 is indicated by the red transitions.

Definition 1 (DTS-sc). A labeled Deterministic Transition System with a stochastic cost function (DTS-sc) is a tuple $T = (S, A, \delta_T, C, AP, L)$, where

- S is a finite set of states,
- A is a finite set of actions,
- $\delta_T : S \times A \rightarrow S$ is a deterministic transition function,
- $C : S \times A \rightarrow \mathcal{D}(\mathbb{R}^N)$ is a stochastic cost function that maps each state-action pair to an N -dimensional probability distribution, where $N \in \mathbb{N}$ and $\mathcal{D}(\mathbb{R}^N)$ is the set of all probability distributions over \mathbb{R}^N ,
- AP is a set of atomic propositions related to the robot tasks, and
- $L : S \rightarrow 2^{AP}$ is a labeling function that assigns to each state $s \in S$ the subset of propositions in AP that are true in s .

A robot plan $\pi \in A^*$ is a finite sequence of actions $\pi = a_0 a_1 \dots a_m$ that induces a trajectory $\tau(s_0, \pi) = s_0 s_1 \dots s_m s_{m+1}$ from a current state $s_0 \in S$, where $s_k \in S$ and, for all $0 \leq k \leq m+1$, $s_{k+1} = \delta_T(s_k, a_k)$. The observation of $\tau(s_0, \pi)$, denoted $L(s_0, \pi)$, is a trace $L(s_0, \pi) = o_0 o_1 \dots o_{m+1}$ where $o_k = L(s_k)$.

Example 2. Consider a simplified DTS-sc model in Fig. 2 of the dishwashing scenario described in Fig. 1 with five states capturing the locations of each dish (J_f : jar on floor, P_r : pitcher on rack, etc.), and two actions load and unload. Each state observes either $\{dry\}$ or $\{wash\}$. Each action from a state takes time and incurs some risk when transporting the jar, modeled with a unique multivariate cost distribution for each state-action pair.

When robot takes action $a \in A$ at state $s \in S$, it endures N distinct types of costs $\mathbf{c}_{s,a} = (c_1, \dots, c_N) \in \mathbb{R}^N$, e.g., energy expenditure and time of execution. This cost vector can generally be stochastic, i.e., the cost value of executing a at s can differ at different instances due to, e.g., changes in the environment or use of randomized algorithms (sampling-based planners). Hence, $\mathbf{c}_{s,a}$ is a random variable whose distribution is given by $\mathcal{C}(s, a)$, i.e., $\mathbf{c}_{s,a} \sim \mathcal{C}(s, a)$. With an abuse of notation, we use \mathbf{c} as the cost random variable for all state-action pairs, and hence $(\mathbf{c} | s, a) = \mathbf{c}_{s,a}$.

We treat $\mathcal{C}(s, a)$ as a conditional distribution of a proba-

bilistic transition-cost generative model, parameterized by the set of parameters Θ , i.e.,

$$\mathcal{C}(s, a) = p(\mathbf{c} | s, a; \Theta).$$

For ease of presentation, we denote (s, a) specific parameters by $\theta_{s,a} \subseteq \Theta$ such that $\Theta = \bigcup_{s \in S, a \in A} \theta_{s,a}$.

We assume robot model T is a Markov process. That is, \mathbf{c} for each state-action pair is independent, i.e., $p(\mathbf{c} | s, a; \theta_{s,a})$ is independent from $p(\mathbf{c} | s', a'; \theta_{s',a'})$ for all $(s, a) \neq (s', a') \in S \times A$. Further, we assume each $p(\mathbf{c} | s, a; \theta_{s,a})$ is a multivariate normal (MVN) distribution $\mathcal{N}(\mu_{s,a}, \Sigma_{s,a})$ parameterized by a non-negative mean $\mu_{s,a} \in \mathbb{R}_{\geq 0}^N$ and covariance $\Sigma_{s,a} \in \mathbb{R}^{N \times N}$, i.e., $\theta_{s,a} = \{\mu_{s,a}, \Sigma_{s,a}\}$. Note that this assumption allows for correlation between types of costs. For instance, if the execution of a robotic action takes a long time, it is likely that it also uses more energy.

In this work, we consider the realistic case where Θ are unknown, i.e., both $\mu_{s,a}$ and $\Sigma_{s,a}$ are unknown for every (s, a) . One of the goals of this work is to learn these parameters from data collected on $\mathbf{c}_{s,a}$ during execution.

Given a plan π and its induced trajectory τ , the cumulative cost of executing π from state s_0 , denoted $\mathbf{C}_{s_0, \pi}$, is the sum of the state-action costs along τ , i.e., $\mathbf{C}_{s_0, \pi} = \sum_{k=0}^m \mathbf{c}_{s_k, a_k}$. Since each \mathbf{c}_{s_k, a_k} is a random variable, $\mathbf{C}_{s_0, \pi}$ is also a random variable, and its distribution is given by the convolution of the \mathbf{c}_{s_k, a_k} distributions, i.e.,

$$p(\mathbf{C} | s_0, \pi; \theta_{s_0, \pi}) = p(\mathbf{c} | s_0, a_0; \Theta) * \dots * p(\mathbf{c} | s_m, a_m; \Theta), \quad (1)$$

where $\theta_{s_0, \pi}$ parameterize the cost distribution of a plan, constructed using Θ . Recall that $p(\mathbf{c} | s_k, a_k; \Theta)$ is assumed to be a MVN; $p(\mathbf{C} | s_0, \pi; \theta_{s_0, \pi})$ is also a MVN. Since true Θ are unknown beforehand, we learn the parameters from experience. We denote the learned parameters with the tilde notation, i.e. $\tilde{\theta}, \tilde{\Theta}$.

B. Robotic Task

The robot is given a complex task that can be completed in finite time. To specify such tasks in a formal manner, we use Linear Temporal Logic over Finite Traces (LTLf) [9], which combines propositional logic with temporal operators.

Definition 2 (LTLf Syntax). A Linear Temporal Logic over Finite Traces (LTLf) formula over AP is recursively defined as

$$\phi := \text{true} \mid o \mid \neg \phi \mid \phi \wedge \phi \mid \bigcirc \phi \mid \phi \mathcal{U} \phi,$$

where $o \in AP$, \neg (negation) and \wedge (conjunction) are Boolean operators, and \bigcirc (next) and \mathcal{U} (until) are temporal operators.

The commonly-used temporal operators “eventually” (\diamond) and “globally” (\square) can be defined as $\diamond \phi \equiv \text{true} \mathcal{U} \phi$ and $\square \phi \equiv \neg \diamond \neg \phi$, respectively.

The semantics of LTLf are defined over finite traces [9]. That is, an LTLf formula ϕ can be satisfied with a finite trace $w \in (2^{AP})^*$ [9], denoted $w \models \phi$. We define the language of ϕ to be the set of finite traces that satisfy ϕ for the first time,

i.e., let $|w|$ be the length of w and $w[k]$ for $1 \leq k \leq |w|$ be the prefix of w with length k ; the *language* of ϕ is defined as

$$\mathcal{L}_\phi = \{w \in (2^{AP})^* \mid w \models \phi \wedge w[k] \not\models \phi \ \forall k < |w|\}.$$

We say plan $\pi \in A^*$ executed from s satisfies ϕ if the resulting trace $L(\pi, s) \in \mathcal{L}_\phi$.

Example 3. Following Example 2, we specify the task $\phi = \diamond(\text{wash} \wedge \diamond(\text{dry}))$ stating “the robot should first wash the dishes, and then dry the dishes.” An example plan that satisfies this task is shown in red in Fig. 2.

We want the robot to execute task ϕ repeatedly. Specifically, we are interested in synthesizing plans that episodically repeat ϕ . That is, once a plan is complete (ϕ is satisfied), a new plan has to be synthesized from the current state to satisfy ϕ again. We assume that such a repetition is always physically possible for the robot, i.e., the robot always ends in a state from which there exists a plan that satisfies ϕ . For instance, in the above example, after drying dishes, the robot always ends in a state from which it can start washing a new pile of dirty dishes. Thus, we focus on the set of satisfying plans from every state, and refer to it as the ϕ -satisfying plan set.

Definition 3 (ϕ -Satisfying Plan Set). Given DTS-sc robot model T , state $s \in S$, and LTLf task formula ϕ , ϕ -Satisfying Plan Set $\Pi(s, \phi)$ is defined as

$$\Pi(s, \phi) = \{\pi \in A^* \mid L(\pi, s) \in \mathcal{L}_\phi\}.$$

C. Optimization Objectives

At each planning (decision) episode from state s , we aim to select a plan $\pi^* \in \Pi(s, \phi)$ that (in addition to completing task ϕ) optimizes for the expected cumulative cost $\mathbb{E}[\mathbf{C} \mid s, \pi; \Theta]$. Since $\mathbf{C}_{s,\pi}$ is N -dimensional, it is in fact an N -objective optimization problem. Moreover, the cost objectives may be competing, i.e., by optimizing one objective, another becomes sub-optimal. Hence, we want to optimize for the trade-offs between the objectives, which is known as Pareto optimal and best defined by the notion of vector dominance.

Definition 4 (Vector Dominance). Let $C, C' \in \mathbb{R}^N$ be two vectors and $C(i)$ be the i -th element of C . Vector C dominates C' , denoted by $C \prec C'$, iff $C(i) \leq C'(i)$ for every $1 \leq i \leq N$ and there exists an i such that $C(i) < C'(i)$.

Definition 5 (Pareto Optimal). For state $s \in S$ and LTLf formula ϕ , plan $\pi \in \Pi(s, \phi)$ is called Pareto optimal if π is non-dominated among all plans in $\Pi(s, \phi)$, i.e.,

$$\nexists \pi' \in \Pi(s, \phi) \quad \text{s.t.} \quad \mathbb{E}[\mathbf{C} \mid s, \pi'; \Theta] \prec \mathbb{E}[\mathbf{C} \mid s, \pi; \Theta].$$

The set of all Pareto optimal plans from s is denoted by $\Pi^*(s, \phi) \subseteq \Pi(s, \phi)$. Then, for each $\pi^* \in \Pi^*(s, \phi)$, $\mathbb{E}[\mathbf{C} \mid s, \pi^*; \Theta]$ is called a Pareto point, and the set of all Pareto points is called the Pareto front, denoted $\Omega(s, \phi)$.

At every planning episode, we desire to select a Pareto optimal plan $\pi^* \in \Pi^*(s, \phi)$ which ensures that the expected cost is not dominated by another plan. However, since Θ are

not available, it is impossible to compute for π^* . Instead, we can use $\tilde{\Theta}$ to compute a learned set of ϕ -satisfying Pareto optimal plans denoted $\tilde{\Pi}^*(s, \phi) \subseteq \Pi(s, \phi)$. Similarly, let $\tilde{\Omega}(s, \phi)$ denote the set of expected cost vectors of each $\pi \in \tilde{\Pi}^*(s, \phi)$.

D. User Preference

There are possibly many Pareto optimal plans that perform ϕ . We are interested in picking a plan π_{pr}^* that achieves the user’s *most-preferred* Pareto point in every episode. Given that our goal is autonomous operation, we require this preference to be provided as an input before deployment. This poses a challenge because the Pareto front is unknown *a priori*, and the user cannot know the exact Pareto point they prefer. Instead, as suggested in recent literature [23, 30], the user can provide a probability density function (pdf) that serves as an expressive way of capturing preferred observed outcomes. We extend this notion to the multi-objective domain, and leverage the expressive pdf preference as a tie-breaker between optimal trade-off candidates.

We specifically focus on the user preference as a normal distribution $p_{\text{pr}}(\mathbf{C}) = \mathcal{N}(\mu_{\text{pr}}, \Sigma_{\text{pr}})$ over the objectives since it can be fully defined by only specifying a mean $\mu_{\text{pr}} \in \mathbb{R}^N$ and covariance $\Sigma_{\text{pr}} \in \mathbb{R}^{N \times N}$, respectively. Thus, by providing μ_{pr} and Σ_{pr} , the user can not only express a desired trade-off region in the objective space, but also specify the willingness to explore alternative trade-offs (Pareto points) that allow the robot to learn more quickly, offering means to address the *exploration-exploitation trade-off* problem. From another perspective, if the user expresses a more *precise* preference (small covariance), the learner can avoid exploring unpreferred trade-offs, often resulting in higher sample efficiency (see Sec. IV). This benefit achieved by eliciting a user’s preference *a priori* is well known to multi-objective optimization [15].

Example 4. Suppose the dish-washing robot described in Fig. 1 is being deployed in a restaurant. If the restaurant is going to be busy, the user may define a preference distribution with mean in a low-time high-risk part of the objective space, whereas if the restaurant will be slow, the user may opt for a high-time, low-risk alternative, both of which can be informed by past observations of how humans wash the dishes. The user may assign the mean based on the costs they expect to receive if they were to perform the task. The covariance can be specified based on their uncertainty given the environment, e.g. a 2- σ estimate of each cost. Under an alternative interpretation, the user can specify a 2- σ ellipse in objective space such that trade-offs that align with the region are prioritized in learning. This specification can be alternatively viewed as outlining a region of interest for multi-objective optimization [15].

To meet the user’s preference, we seek a Pareto point that is not “surprising.” The information theoretic definition of *surprise* is the negative log of model evidence [31]. Similarly, we define the surprise of an observation obtained by following a plan π as:

$$\text{Surprise}(\pi) = -\log p(\mathbf{C} \mid \pi). \quad (2)$$

This quantity can be derived by marginalizing the following generative model:

$$\begin{aligned} (2) &= -\log \int_s \int_{\tilde{\theta}_{s,\pi}} p(s, \tilde{\theta}_{s,\pi}, \mathbf{C} | \pi) d\tilde{\theta}_{s,\pi} ds, \\ &= -\log \int_s \int_{\tilde{\theta}_{s,\pi}} p(s, \tilde{\theta}_{s,\pi} | \mathbf{C}, \pi) p(\mathbf{C} | \pi) d\tilde{\theta}_{s,\pi} ds. \end{aligned} \quad (3)$$

In active inference (AIF), one can inject a bias towards seeing certain outcomes by substituting the predictive outcome distribution $p(\mathbf{C} | \pi)$ with the prior preference $p_{\text{pr}}(\mathbf{C})$ (a.k.a. “evolutionary prior”) that is independent of any plan [22, 32]. We refer to the biased surprise as *surprise w.r.t. p_{pr}* :

$$\text{Surprise}(\pi, p_{\text{pr}}) = -\log \int_s \int_{\tilde{\theta}_{s,\pi}} p(s, \tilde{\theta}_{s,\pi} | \mathbf{C}, \pi) p_{\text{pr}}(\mathbf{C}) d\tilde{\theta}_{s,\pi} ds. \quad (4)$$

In essence, under AIF, an agent acts to see desired observations. An overview of AIF is provided in Sec. V-B.

E. Problem Statement

The problem we consider is as follows.

Problem 1. Consider a DTS-sc robot model T with unknown cost distributions, an LTLf task specification ϕ that the robot is to repeatedly complete, and a user preference distribution $p_{\text{pr}}(\mathbf{C})$ over N cost objectives. Let $K \in \mathbb{N}$ denote the episode, i.e. the number of times the robot has already completed task ϕ , and $s_K \in S$ be the start state of the robot in the K -th episode. Then, for each K , synthesize a ϕ -satisfying plan $\pi_K \in \Pi(s_K, \phi)$ such that

- (i) as $K \rightarrow \infty$, π_K becomes Pareto optimal, i.e., $\pi_K \in \Pi^*(s_K, \phi)$, and
- (ii) π_K minimizes surprise in (4) among plans in $\tilde{\Pi}^*(s_K, \phi)$, i.e., $\pi_K \in \arg\min_{\pi \in \tilde{\Pi}^*(s_K, \phi)} \text{Surprise}(\pi, p_{\text{pr}})$,
- (iii) as $K \rightarrow \infty$, the resulting Pareto front $\tilde{\Omega}(s_K, \phi)$ converges to the true Pareto front $\Omega(s_K, \phi)$.

Note that, in Problem 1, satisfaction of ϕ (qualitative objective) at each episode is a hard constraint, and the optimization (quantitative) objectives in (i)-(iii) are soft constraints. Hence, by incorporating safety requirements in ϕ , safety satisfaction is guaranteed throughout the process. Objectives (ii) and (iii) can be juxtaposed as *exploitation* and *exploration* respectively, i.e., the robot should converge to the preferred trade-off but must explore the transition cost model to learn other alternatives on the Pareto front. Achieving an intelligent balance between exploitation and exploration is necessary in order to effectively learn Pareto-optimal ways of satisfying the task in as few episodes as possible.

In our approach, we address this challenge by using free energy minimization to naturally balance exploitation and exploration over the Pareto front.

III. APPROACH

Our approach to Problem 1 is an iterative MORL framework with four phases: 1) *planning* Pareto optimal plans, 2) *selection* of the preferred Pareto optimal plan, 3) *execution* of the

selected plan, and 4) *update* to the executed state-action cost parameters, as illustrated in Fig. 3. *Planning*, *selection*, and *update* are all dependent on the learning status of the robot. Each episode, the planner computes an approximate set of Pareto-optimal plans with respect to $\tilde{\Theta}$, which is then passed to the selection layer, responsible for choosing the preferred optimal trade-off. The preferred trade-off is executed on the robot, and finally, $\tilde{\Theta}$ is updated with cost measurements collected during execution. We start by formalizing our approach to modeling $\tilde{\Theta}$.

A. Learning the Transition Cost Model

Recall that the cost $\mathbf{c}_{s,a}$ of each (s, a) is distributed as $p(\mathbf{c} | s, a; \theta_{s,a}) = \mathcal{N}(\mu_{s,a}, \Sigma_{s,a})$. However, $\theta_{s,a}$ are unknown and must be learned using experience. We take a Bayesian approach to learning $\theta_{s,a}$ by modeling with random variable $\tilde{\theta}_{s,a}$. Specifically, we assume that the robot collects a realized sample $\hat{\mathbf{c}}$ drawn from $p(\mathbf{c} | s, a; \theta_{s,a})$ each time a is executed from s . Let $D_{s,a} = \{\hat{\mathbf{c}}_1, \dots, \hat{\mathbf{c}}_l\}$ be the set of l realizations of $\mathbf{c}_{s,a}$. We maintain a belief over $\tilde{\theta}_{s,a}$ using the Normal Inverse-Wishart (NIW) distribution, i.e.,

$$p(\tilde{\theta}_{s,a}) = \text{NIW}_{s,a}(\lambda_0, \kappa_0, \Lambda_0, \nu_0).$$

Note that since $\tilde{\theta}_{s,a}$ parameterize the cost distributions, $p(\tilde{\theta}_{s,a})$ is effectively a distribution *over* a family of distributions. The NIW prior is chosen based on two highly beneficial properties: i) the posterior can be analytically represented as a parameterized distribution, and ii) the posterior belongs to the same distribution family as the prior (also NIW) [33]. Consequently, we can write the posterior as another NIW distribution with updated parameters,

$$p(\tilde{\theta}_{s,a} | D_{s,a}) = \text{NIW}_{s,a}(\lambda, \kappa, \Lambda, \nu).$$

Since the prior and posterior beliefs share the same distributional family, subsequent analysis of the statistics of $\tilde{\theta}_{s,a}$ does not depend on whether or not a prior or posterior distribution is used, i.e. if data has been collected for that state action distribution or not. Note that some works assume the covariance is known *a priori* [34], limiting the fidelity of the model. NIW strikes a balance between an over simplified parametric model (known covariance) and a sample inefficient non-parametric model [35]. We detail the the update procedure in Sec. III-D. If the user has an informed guess about the mean and covariance of a certain state-action distribution, they can set λ_0 and Λ_0 respectively and increase the κ_0 and ν_0 , reducing the number of episodes required to learn $\tilde{\Theta}$.

B. Planning

At the start of each planning episode, we employ a multi-objective task planner to compute the set of approximate Pareto-optimal plans $\tilde{\Pi}^*(s_K, \phi)$ described in Def. 5. To this end, we formulate a *correct-by-construction* multi-objective shortest-path graph search problem, similar to [10], which can be solved with the following procedure.

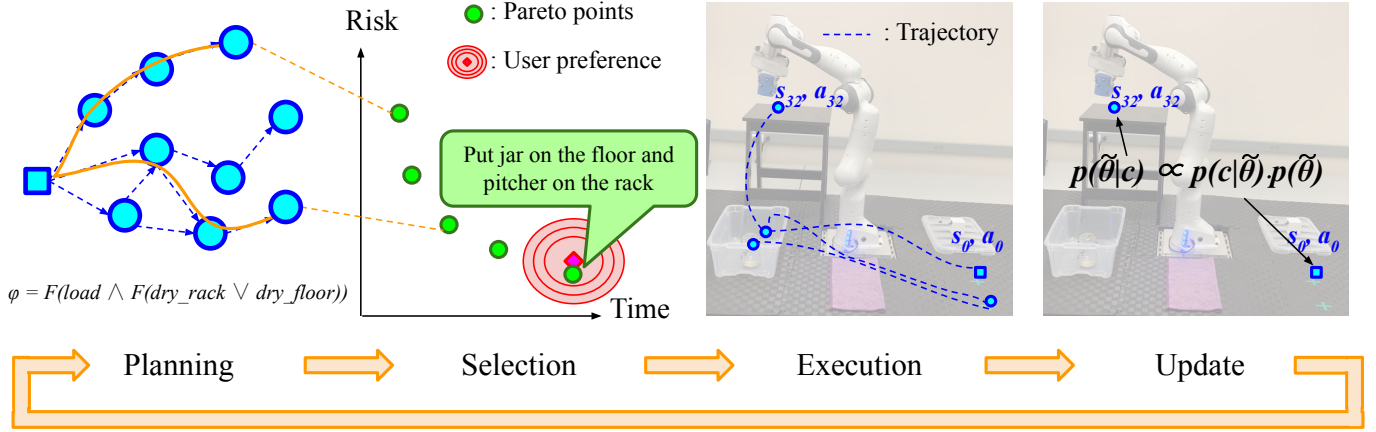


Fig. 3: Proposed multi-objective safe reinforcement learning architecture: Four phases are repeatedly executed. The robot gradually learns the state-action cost parameters and selects the user preferred optimal task plan. Planning (1) computes a set of Pareto-optimal plans. Selection (2) chooses the user’s preferred trade-off from the Pareto-optimal plans. The preferred trade-off is executed (3), and the cost model is updated (4) with measurements collected during execution.

1) *Product Graph Construction*: An LTLf formula can be translated into a deterministic finite automaton (DFA) [9], a finite state-machine that precisely captures the language of ϕ .

Definition 6 (Deterministic Finite Automaton). A DFA constructed from LTLf formula ϕ is a tuple $Q_\phi = (\Gamma, \gamma_0, W, \delta_{Q_\phi}, \mathcal{F})$ where Γ is a set of states, $\gamma_0 \in \Gamma$ is the initial state, $W = 2^{AP}$ is the alphabet, $\delta_{Q_\phi} : \Gamma \times W \rightarrow \Gamma$ is a deterministic transition function, and \mathcal{F} is a set of accepting states.

A trace $w = w_0 \dots w_{m-1} \in (2^{AP})^*$ induces a finite run $\tau^{Q_\phi} = \gamma_0 \gamma_1 \dots \gamma_m$ on Q_ϕ , where $\gamma_{k+1} = \delta_{Q_\phi}(\gamma_k, w_k)$. Run τ^{Q_ϕ} is called accepting if $\gamma_m \in \mathcal{F}$. If τ^{Q_ϕ} is accepting, then trace w is accepted by Q_ϕ and satisfies ϕ , i.e., $w \models \phi$.

Using T to capture the physical capability of the robotic system, along with Q_ϕ to capture the temporal attributes of ϕ , we can construct a product automaton that intersects the restrictions of both T and Q_ϕ .

Definition 7 (Product Automaton). Given a DTS-sc $T = (S, A, \delta_T, \mathcal{C}, AP, L)$, current state $s_K \in S$, and DFA Q_ϕ , a Product Automaton is a tuple $\mathcal{P} = (P, \rho_K, A, \delta_P, \mathbf{v}, \mathcal{F}_P)$, where

- $P = S \times \Gamma$ is a set of states,
- $\rho_K = (s_K, \gamma'_0)$, where $\gamma'_0 = \delta_{Q_\phi}(\gamma_0, L(s_K))$, is the starting state in the current episode,
- A is the same action set in T ,
- $\delta_P : P \times A \rightarrow P$ is a transition function, where for states $\rho = (s, \gamma)$ and $\rho' = (s', \gamma')$ and action $a \in A$, transition $\rho' = \delta_P(\rho, a)$ exists if $s' = \delta_T(s, a)$ and $\gamma' = \delta_{Q_\phi}(\gamma, L(s'))$,
- $\mathbf{v} : P \times A \rightarrow \mathbb{R}_{\geq 0}^N$, is a transition cost-vector function, and
- $\mathcal{F}_P = S \times \mathcal{F}$ is a set of accepting states.

Note that unlike stochastic cost function \mathcal{C} in T , cost-vector function $\mathbf{v}(\rho, a) \in \mathbb{R}_{\geq 0}^N$ is not a distribution. We elaborate on \mathbf{v} further below.

Similar to T , a plan π induces a run $\tau^P = \rho_0 \dots \rho_{m+1}$ on \mathcal{P} where $\rho_{k+1} = \delta_P(\rho_k, a_k)$. Run τ^P is accepting iff $\rho_{m+1} \in$

\mathcal{F}_P . Therefore, by construction, the set of ϕ -satisfying plans $\Pi(s_K, \phi)$ is equal to the set of all plans that induce paths on \mathcal{P} from ρ_K to a $\rho \in \mathcal{F}_P$.

2) *Transition Cost-Vector Function*: We aim to define cost-vector function \mathbf{v} and accordingly formulate a multi-objective graph search problem on \mathcal{P} that enables the computation of the set of Pareto optimal ϕ -satisfying plans $\Pi^*(s_K, \phi)$. Consider current state $\rho_K = (s_K, \gamma_0)$, plan $\pi \in \Pi(s_K, \phi)$, and induced path $\tau(\rho_K, \pi) = \rho_{K,0} \dots \rho_{K,m+1}$. Ideally, for every $\rho = (s, \gamma) \in P$ and $a \in A$, we want $\mathbf{v}((s, \gamma), a) = \mathbb{E}[c|s, a; \Theta]$ because the total cost of induced path $\tau^P(\rho_K, \pi)$ becomes

$$\sum_{i=0}^{m+1} \mathbf{v}(\rho_{K,i}, \pi_i) = \sum_{i=0}^{m+1} \mathbb{E}[c|s_{K,i}, \pi_i; \Theta] = \mathbb{E}[C|s_K, \pi; \theta_{s_K, \pi}].$$

Therefore, by assigning $\mathbf{v}(\rho, a) = \mathbb{E}[c|s, a; \Theta]$, the multi-objective Pareto front of the graph search problem on P is exactly equal to $\Omega(s_K, \phi)$ (see Def. 5). Then, existing algorithms such as a multi-objective variant of Dijkstra’s algorithm or A^* [36] can be employed to compute Ω .

We note that, in this work, we consider deterministic plans instead of stochastic policies, since the optimization is performed for only the next task completion. Stochastic policies would generate a continuous Pareto front, i.e., the convex hull of $\Omega(s_K, \phi)$, however points on the continuous Pareto front may only be achievable *in expectation* instead of a single execution of the task. We recognize that for batch task completion scenarios, extending our approach to using a continuous Pareto front and stochastic policies may be a valuable future direction.

Even though the Pareto front is guaranteed to be finite since the plans are deterministic and the task is finite-time satisfiable, multi-objective graph search algorithms are exponential in the worst case. However, the average branching factor for a N -dimensional vector belonging to a set of size \mathcal{G} , i.e., the set of all unique cost vector edge weights, is reduced to $O((\log |\mathcal{G}|)^{N-1})$, with certain conditions on the ordering [36].

Recall that Θ is unknown. A naive approach simply uses

learned estimates $\mathbb{E}[\tilde{\Theta}]$ for \mathbf{v} instead, but that suffers from biasing the solution towards taking transitions that have low estimated cost, regardless of how accurate $\tilde{\Theta}$ is. To address this, we interweave exploration into the graph search problem, as detailed below.

3) *Pareto-Regret*: The efficacy of learning can be analyzed with the notion of cumulative ‘‘Pareto-regret’’, quantifying how sub-optimal each plan is with respect to the true Pareto front $\Omega(s_K, \phi)$. Formally, given π with true mean cumulative cost $\mu_{s_K, \pi} = \mathbb{E}[\mathbf{C}|s_K, \pi; \Theta]$, the Pareto regret for a given episode is defined as

$$r(s_K, \pi) = \min_{\epsilon} \{ \mu_{s_K, \pi} - \epsilon \mathbf{1} \mid \mu^* \not\prec \mu_{s_K, \pi} - \epsilon \mathbf{1} \quad \forall \mu^* \in \Omega(s_K, \phi) \}, \quad (5)$$

where $\mathbf{1}$ is a vector of ones with appropriate dimension. Note that if $\pi \in \Pi^*(s_K, \phi)$, then $r(s_K, \pi) = 0$. The cumulative regret up to episode K is simply the summed regret for every episode, i.e. $R = \sum_{i=0}^K r(s_i, \pi_i)$. To minimize cumulative regret, we augment the transition cost-vector with an exploration strategy.

4) *Pareto Cost-LCB*: We adapt the well established Pareto-UCB1 [16, 37] strategy for computing an estimate Pareto front that balances the current best estimate cost model with a bonus reduction in cost. The learned mean transition-cost is calculated using the estimated parameters $\mathbb{E}[\tilde{\theta}_{s,a}]$. To maintain optimality guarantees of the aforementioned graph search method, we rectify each element of \mathbf{v} to be non-negative. The cost-lower confidence bound (LCB) cost vector $\mathbf{v}(\rho, a)$ is computed element-wise as:

$$v_i(\rho, a) = \max \left\{ 0, \mathbb{E}[\mathbf{c}|s, a; \mathbb{E}[\tilde{\theta}_{s,a}]]_i - \alpha \sqrt{\log(k_g)/n(s, a)} \right\}$$

where α is a ‘‘confidence’’ hyperparameter, k_g is the current global time step across all episodes, and $n(s, a)$ is the number of times action a has been executed from state s .

Note that many multi-objective multi-armed bandit confidence bound approaches provide theoretical regret bounds that are logarithmic with respect to the number of episodes. It is not straight forward to extend these regret bounds to this framework due to the exponential nature of the planning space $\Pi(s_K, \phi)$ which has size $O(|A|^{|P|})$. Hence, we leave theoretical regret bounds to future research.

Using the techniques described above, we can now compute the approximate set of Pareto optimal plans $\tilde{\Pi}^*(s_K, \phi)$ while accounting for exploration of the environment. The exploration in *planning* aids in the effort in addressing goals (ii) and (iii) of Problem 1, however, to uphold proper learning of the Pareto front (iii), the selection must also explore among trade-offs.

C. Pareto Point Selection

For the *selection* problem, one can think of candidate plans $\pi \in \tilde{\Pi}^*(s_K, \phi)$ computed by the planner as abstract macro-actions, such that high-level decision making can be done sequentially for each episode. Fig. 4 provides a visual representation of the abstracted selection problem with respect to the evolution of the robotic system over each episode.

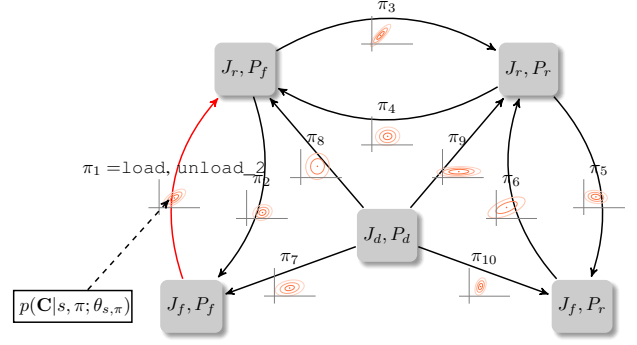


Fig. 4: By treating plans π as macro-actions, the *selection* decision making problem becomes sequential. The example plan seen in red in Fig. 2 can now be simply represented by a single state-macro-action transition embedded into π_1 (red) as shown.

As described in Sec. II-D, ideally, the agent should select an action minimizing the surprise of outcomes. However, since it is analytically intractable to marginalize the joint distribution to derive the surprise, its upper bound, (free energy) is minimized. Yet, the outcomes \mathbf{C} cannot be observed until a plan π is actually executed, thus, the agents end up minimizing the so-called expected variational free energy (EFE) in active inference based decision-making. In this Pareto point selection problem, the EFE is described as follows (see Appendix ?? for detailed derivation).

$$\begin{aligned} \text{EFE}(s_K, \pi^*, p_{\text{pr}}) = & -\mathbb{E}_{q(\mathbf{C}|\pi^*)} \left[\log p_{\text{pr}}(\mathbf{C}) \right] \\ & - \mathbb{E}_{q(\mathbf{C}|\pi^*)} \left[D_{KL}(q(s_K|\mathbf{C}, \pi^*) || q(s_K|\pi^*)) \right] \\ & - \mathbb{E}_{q(\mathbf{C}|\pi^*)} \left[D_{KL}(q(\tilde{\theta}_{s_K, \pi^*} | \mathbf{C}, s_K, \pi^*) || q(\tilde{\theta}_{s_K, \pi^*} | s_K, \pi^*)) \right], \end{aligned} \quad (6)$$

where $q(\cdot)$ represents a proposal distribution. EFE is a variational quantity [38], yielding freedom in the choice of proposal distribution $q(\tilde{\theta}_{s_K, \pi} | s_K, \pi^*)$. As $q(\tilde{\theta}_{s_K, \pi} | s_K, \pi^*)$ more closely resembles $p(\tilde{\theta}_{s_K, \pi} | \mathbf{C}, s_K, \pi^*)$, the upper bound on (expected) surprise decreases. The first term of EFE in (6) represents how much the predicted cost distribution $q(\mathbf{C}|\pi^*)$ aligns with $p_{\text{pr}}(\mathbf{C})$ (i.e. exploitation), and the second and third terms represent how much the uncertainties of s_K and $\tilde{\theta}_{s_K, \pi^*}$ can be reduced by following π^* and measuring \mathbf{C} (i.e. exploration).

Since the user does not know the true Pareto-optimal region *a priori*, EFE cannot be used to scalarize any plan in Π , as it may often be sub-optimal. Therefore, as described in Sec. V-A, the preferred plan $\pi_{\text{pr}} = \arg\min_{\pi \in \Pi} \text{EFE}(\pi, p_{\text{pr}})$ is not guaranteed to also be an element in Π^* and vice versa. Consequently, in order to ensure that the user’s preference does not bias the system away from optimal behavior, this preference is only expressed over Pareto-optimal candidates. So, $\Pi^*(s_K, \phi)$ is scalarized via EFE in order to adhere to p_{pr} and reduce uncertainties in the belief over $\tilde{\theta}_{s_K, \pi^*}$. Since T models deterministic transitions, the second term of (6) can

be ignored. Expanding the third term of (6) yields

$$\begin{aligned} \text{EFE}(s_K, \pi^*, p_{\text{pr}}) &= -\mathbb{E}_{q(\mathbf{C}|\pi^*)} \left[\log p_{\text{pr}}(\mathbf{C}) \right] \\ &\quad - \mathbf{H}[q(\tilde{\theta}_{s_K, \pi} | s_K, \pi^*)] + \mathbb{E}_{q(\mathbf{C}|\pi^*)} \left[\mathbf{H}[q(\tilde{\theta}_{s_K, \pi^*} | \mathbf{C}, s_K, \pi^*)] \right], \end{aligned} \quad (7)$$

where $\mathbf{H}[\cdot]$ represents the entropy of a pdf. Then, the preferred plan is

$$\pi_{\text{pr}}^* = \underset{\pi^* \in \Pi^*(s_K)}{\text{argmin}} \text{EFE}(s_K, \pi^*, p_{\text{pr}}). \quad (8)$$

Due to the Markov property of DTS-sc and (1), μ_{s_K, π^*} and Σ_{s_K, π^*} of the convolved distribution $p(\mathbf{C}|s_K, \pi^*; \theta_{s_K, \pi^*})$ are parameterized by $\sum_k \mu_{s_k, a_k}$ and $\sum_k \Sigma_{s_k, a_k}$ of each cost distribution respectively. According to the estimate transition cost model, $\tilde{\theta}_{s, a} = \{\tilde{\mu}_{s, a}, \tilde{\Sigma}_{s, a}\}$ are random variables themselves; therefore,

$$p(\tilde{\theta}_{s_K, \pi^*} | s_K, \pi^*) = p(\tilde{\theta}_{s_0, a_0}) * \dots * p(\tilde{\theta}_{s_m, a_m}). \quad (9)$$

As described in Section III-A, each $p(\tilde{\theta}_{s, a})$ is a unique NIW, which makes (9) analytically intractable. To enable computation for π_{pr}^* in (8), we leverage both the functional freedom in choosing proposal distributions $q(\cdot)$ as well as three statistical approximations of (7): 1) certainty equivalence of the predicted observation distributions [39], 2) central limit theorem (CLT) [40], 3) and Monte Carlo sampling [41].

1) *Approximating the First Term of (7)*: Recall $p(\tilde{\theta}_{s_K, \pi^*} | s_K, \pi^*)$ is the belief over parameters of the predicted observation distribution $q(\mathbf{C}|\pi^*)$. To address the intractability, we can extract the current estimate parameters $\mathbb{E}[\tilde{\theta}_{s_K, \pi^*}]$. This certainty-equivalence approximation makes $q(\mathbf{C}|\pi^*)$ a simple multivariate normal distribution

$$\begin{aligned} q(\mathbf{C}|\pi^*) &= \int_{\tilde{\theta}_{s_K, \pi^*}} q(\mathbf{C}, \tilde{\theta}_{s_K, \pi^*} | \pi^*) d\tilde{\theta}_{s_K, \pi^*} \\ &\approx q(\mathbf{C}|\pi^*; \mathbb{E}[\tilde{\theta}_{s_K, \pi^*}]). \end{aligned} \quad (10)$$

The first term can then be calculated as follows

$$\begin{aligned} -\mathbb{E}_{q(\mathbf{C}|\pi^*)} \left[\log p_{\text{pr}}(\mathbf{C}) \right] &\approx \log((2\pi)^{-k/2} |\Sigma_{\text{pr}}|^{-1/2}) \\ &\quad + \frac{1}{2} (\mu_{\text{pr}}^T \Sigma_{\text{pr}}^{-1} \mu_{\text{pr}} + \text{tr}(\Sigma_{\text{pr}}^{-1} \Sigma_{s_K, \pi^*}) + \mu_{s_K, \pi^*}^T \Sigma_{\text{pr}}^{-1} \mu_{s_K, \pi^*}) \\ &\quad - \mu_{\text{pr}}^T \Sigma_{\text{pr}}^{-1} \mu_{s_K, \pi^*}, \end{aligned} \quad (11)$$

where μ_{pr} and Σ_{pr} are the mean vector and the covariance matrix of the user's preference distribution (see Appendix ?? for detailed derivation).

2) *Calculating the Second Term of (7)*: The convolved distribution $p(\tilde{\theta}_{s_K, \pi^*} | s_K, \pi^*)$ is analytically intractable. However, we can leverage the CLT to determine a suitable proposal distribution $q(\cdot)$ that closely models $p(\cdot)$. Since each $p(\tilde{\theta}_{s, a})$ is NIW-distributed, both the mean and variance are well-defined when $\kappa > 0$ and $\nu > N + 1$. Therefore, under the CLT, $p(\tilde{\theta}_{s_K, \pi^*} | s_K, \pi^*)$ tends towards a MVN distribution as the plan length $|\pi^*|$ becomes large [42].

Consider the following vectorization of the parameters $\theta_{s, a}$

$$\text{vec}(\theta_{s, a}) = (\mu_1, \dots, \mu_N, \sigma_{1,1}^2, \dots, \sigma_{N,N}^2), \quad (12)$$

where μ_1, \dots, μ_N represent the components of $\mu_{s, a}$, and $\sigma_{1,1}^2, \dots, \sigma_{N,N}^2$ are the unique upper triangular elements of

$\Sigma_{s, a}$. In total, the vectorization has dimension $N(N + 3)/2$. Due to the linearity of expectation and independence between state-action pairs,

$$\mathbb{E}[\tilde{\theta}_{s_K, \pi}] = \sum_{k=0}^m \mathbb{E}[\tilde{\theta}_{s_k, a_k}], \quad (13)$$

$$\text{var}[\tilde{\theta}_{s_K, \pi}] = \sum_{k=0}^m \text{var}[\tilde{\theta}_{s_k, a_k}]. \quad (14)$$

Therefore, the distribution $p(\tilde{\theta}_{s_K, \pi^*} | s_K, \pi^*)$ can be approximately represented by multivariate normal proposal distribution parameterized by the vectorized cumulative mean and variance

$$q(\tilde{\theta}_{s_K, \pi^*} | s_K, \pi^*) \equiv \mathcal{N} \left(\sum_k \mathbb{E}[\text{vec}(\tilde{\theta}_{s_k, a_k})], \sum_k \text{var}[\text{vec}(\tilde{\theta}_{s_k, a_k})] \right). \quad (15)$$

Since $q(\tilde{\theta}_{s_K, \pi^*} | s_K, \pi^*)$ is MVN, the entropy of the proposed distribution (second term) can be represented analytically rewritten as follows.

$$\begin{aligned} \mathbf{H}[q(\tilde{\theta}_{s_K, \pi} | s_K, \pi^*)] &= \frac{1}{2} \log \left(\det \left(\sum_k \text{var}[\text{vec}(\tilde{\theta}_{s_k, a_k})] \right) \right) \\ &\quad + \frac{N(N + 3)}{4} (1 + \log 2\pi). \end{aligned} \quad (16)$$

3) *Approximating the Third Term of (7)*: Using both the certainty equivalence approximation in (10) and the CLT approximation in (15), the remaining third term can be approximated using Monte Carlo sampling of the expectation. To sample the expectation, a cumulative cost vector must be sampled from $q(\mathbf{C}|\pi^*)$. Using (10), this sample (denoted $\hat{\mathbf{c}}$) can be drawn from the MVN $\hat{\mathbf{C}} \sim q(\mathbf{C}|\pi^*; \mathbb{E}[\tilde{\theta}_{s_K, \pi^*}])$. However, note that the posterior $p(\tilde{\theta}_{s_K, \pi^*} | \mathbf{C}, s_K, \pi^*)$ is itself a convolved NIW conditioned on \mathbf{c} . To construct an analytical approximation of the posterior, we must instead expand the sampling procedure to each state-action distribution. Instead of sampling $\hat{\mathbf{C}}$ directly from $q(\mathbf{C}|\pi^*; \mathbb{E}[\tilde{\theta}_{s_K, \pi^*}])$, $\hat{\mathbf{c}}$ can instead be constructed from the observation distribution for each state action pair

$$\hat{\mathbf{C}} = \sum_k \hat{\mathbf{c}}_{s_k, a_k}, \quad (17)$$

where $\hat{\mathbf{c}}_{s_k, a_k} \sim p(\mathbf{c}|s_k, a_k; \mathbb{E}[\tilde{\theta}_{s_k, a_k}])$. For each sample $\hat{\mathbf{c}}_{s_k, a_k}$, the respective posterior $p(\tilde{\theta}_{s_k, a_k} | \hat{\mathbf{c}}_{s_k, a_k})$ are computed using the equations found in Sec. III-D. The distribution $p(\tilde{\theta}_{s_k, a_k} | \hat{\mathbf{c}}_{s_k, a_k})$ is the conjugate NIW posterior. Therefore, using the same process described in Sec. III-C2, an approximation of $\mathbf{H}[q(\tilde{\theta}_{s_K, \pi} | \hat{\mathbf{C}}, s_K, \pi^*)]$ can be calculated. This sampling procedure is repeated n_s times to compute the third term of (7). Using the aforementioned techniques, we can optimize for π_{pr}^* in (8). While this sampling procedure is computationally burdensome, the selection procedure runs in $O(n_s m |\bar{\Pi}^*|)$, linear in the size of number of candidate Pareto optimal plans.

Note that unlike the prior distribution in the second term of (7), the proposal posterior $q(\tilde{\theta}_{s_K, \pi^*} | \mathbf{C}, s_K, \pi^*)$ is not a variational proposal distribution. In fact, the approximation of true posterior $p(\tilde{\theta}_{s_K, \pi^*} | \mathbf{C}, s_K, \pi^*)$ with $q(\tilde{\theta}_{s_K, \pi^*} | \mathbf{C}, s_K, \pi^*)$

is an approximation *inherent* to active inference [32], as $p(\tilde{\theta}_{s_K, \pi^*} | \mathbf{C}, s_K, \pi^*)$ is often not analytically tractable. This approximation intuitively relies on the assumption that the internal model of the agent is accurate enough to well predict the true posterior. Our empirical evaluations in Appendix ?? show that the approximation of convolved NIW with convolved MVN agrees with the CLT. That is, the approximation is fairly accurate for the typical plan length seen in the experiments in Sec. IV, and increases as the plan length increases (Fig. ??) and more data is collected (Fig. ??). Additionally, refer to Appendix ?? for an error analysis of the Monte Carlo sampling procedure described in Sec. III-C3 with respect to the number of samples n_s , plan length, and collected data.

D. Execution and Parameter Update

After obtaining the preferred optimal trade-off plan π_{pr}^* , it is executed on the robotic platform (or in simulation). During the execution, measurements for the cost of each state-action \hat{c} are collected. Recall, the cost parameters are distributed as $p(\tilde{\theta}_{s,a} | D_{s,a}) = NIW_{s,a}(\lambda, \kappa, \Lambda, \nu)$. Given a set of measurements $D_{s,a} = \{\hat{c}_0, \dots, \hat{c}_l\}$, the conjugate posterior is a NIW parameterized as follows.

$$\lambda = \frac{\kappa_0 \lambda_0 + l \bar{c}}{\kappa_0 + l}, \quad (18)$$

$$\kappa = \kappa_0 + l, \quad (19)$$

$$\Lambda = \Lambda_0 + \frac{\kappa_0 l}{\kappa_0 + l} (\bar{c} - \lambda_0) (\bar{c} - \lambda_0)^T + \quad (20)$$

$$\sum_{j=0}^l (\hat{c}_j - \bar{c}) (\hat{c}_j - \bar{c})^T, \quad (21)$$

$$\nu = \nu_0 + l,$$

By iteratively performing each described phase, the agent can safely complete the task, guided by the user's preference. Through the iterative combination of exploratory planning, surprised-based selection, execution, and Bayesian update of cost distributions, the proposed MORL framework can achieve all three goals described in Prob. 1. Now, we evaluate the empirical efficacy through experiments.

IV. EXPERIMENTS

To evaluate the effectiveness of our MORL framework for autonomous robotic decision making in unknown environments, we performed three case studies: (i) an illustrative simulation study, (ii) two numerical benchmarking experiments for comparison against the state-of-the-art, and (iii) a hardware experiment to demonstrate the real-world applicability of the method¹.

Numerically, we aim to assess (i) how close the selected plan π_{pr}^* is to being Pareto optimal, and (ii) how well Π^* is represented by $\tilde{\Pi}^*$. Metric *cumulative Pareto-regret* directly evaluates (i). To properly evaluate (ii), we define *Pareto-bias*

metric for a true Pareto front Ω and an estimate Pareto front $\tilde{\Omega}$ as

$$B = \frac{1}{|\tilde{\Omega}|} \sum_{\mathbf{C}_i \in \tilde{\Omega}} \min_{\mathbf{C}_j \in \Omega} (d(\mathbf{C}_i, \mathbf{C}_j)) + \frac{1}{|\Omega|} \sum_{\mathbf{C}_j \in \Omega} \min_{\mathbf{C}_i \in \tilde{\Omega}} (d(\mathbf{C}_j, \mathbf{C}_i)), \quad (22)$$

where

$$d(\mathbf{C}_i, \mathbf{C}_j) = \mathcal{W}_2(p(\mathbf{C}_i | s_K, \pi_i; \theta_{s_K, \pi_i}), q(\mathbf{C}_j | s_K, \pi_j; \mathbb{E}[\tilde{\theta}_{s_K, \pi_j}]))$$

is the Wasserstein-2 (\mathcal{W}_2) distance. The first term in (22) quantifies the estimate Pareto front's coverage of the true Pareto front, and the second term penalizes outlier/excess trade-offs in the estimate Pareto front. If each true Pareto point is exactly similar to an estimate Pareto point, and visa versa, then $B = 0$, otherwise $B > 0$.

A. Simulated Mars Surface Exploration Study

1) *Motivation*: Suppose a Mars rover is tasked to collect scientifically interesting minerals from target sample sites (orange region in Fig. 5), designated based on past mission data [43]. Due to the limited number of sample tubes, it is required to go back and forth between the deposit location (green region) and the target site. Since the closer sampling region is in the sun, the rover must trade-off between low-time, high-radiation for efficient science data collection, or high-time, low-radiation to avoid long-term damage. Therefore, to protect itself while gaining sufficient amount of science, it must strike a balance between the two. Moreover, although the rover knows the qualitative information about the environment (e.g., where sand, base, washing station, etc. are located), it is still necessary to estimate online the time and radiation cost of collecting a sample. Informed by past mission data and scientific expertise, the user can tune an appropriate p_{pr} .

2) *Simulation Setup*: For the sake of simplicity, the rover can travel in cardinal directions, and may not move through obstacles (black). The task specification is given as

$$\phi_{\text{MR}} = \diamond(\text{sample} \wedge \diamond(\text{deposit})) \wedge \square(\text{sand} \rightarrow (\neg \text{base } \mathcal{U} \text{ wash}));$$

in English, “collect a sample, then deposit it, and, if sand is visited then wash before returning to base.” Each transition takes time (objective 1) measured in minutes, and transitions in the *sun* accumulate harmful radiation (objective 2) measured in micrograys. The true mean costs for transitions in each region are given as follows:

$$\begin{aligned} \mu_{\text{sample}(\text{left})} &= (6, 16) && \text{(in the sun),} \\ \mu_{\text{sample}(\text{right})} &= (6, 0) && \text{(out of the sun),} \\ \mu_{\text{sand}} &= (3, 7) && \text{(slows the rover in the sun),} \\ \mu_{\text{wash}} &= (11, 31) && \text{(long duration of radiation exposure),} \end{aligned}$$

$\mu_{\text{sun}} = (1, 1)$, and all other transitions have $\mu = (1, 0)$. Covariance values are omitted for brevity. The user informs the prior transition cost for all transitions to be $\lambda_0 = (0.5, 0)$. For the longevity of the rover, the user aims to endure a small amount of radiation and prefers that the mission is completed

¹To see a video of the simulation and hardware experiments, visit <https://youtu.be/tCRJwqeT-f4>

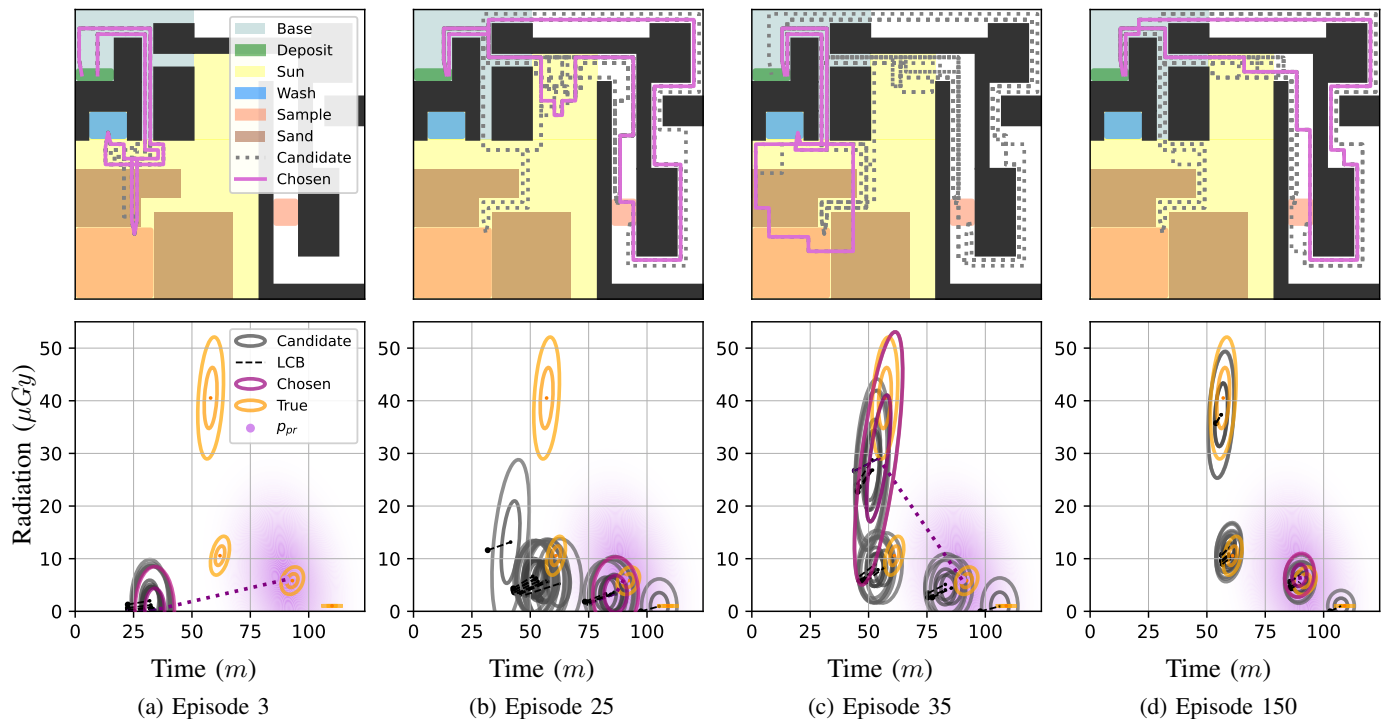


Fig. 5: Simulated rover sample collection: A rover is tasked with repetitively collecting a *sample* and delivering it to *deposit*. Moving takes time and collects radiation (in the sun). The user prefers that the sample is collected in roughly 90 minutes with about $6 \mu Gy$ of radiation, represented with p_{pr} . Each computed plan (top row) corresponds to a Pareto point (bottom row) showing the estimated cost distribution with 1 & 2- σ ellipses. The selected Pareto point and corresponding plan is shown in purple, with a dotted line from the mean of the preference distribution to the chosen distribution. Top row figures show the computed plans, and the bottom row figures show the estimated Pareto points. The true optimal trade-off plans are shown for comparison in four episode snapshots. *Video*: <https://youtu.be/tCRJwqeT-f4>.

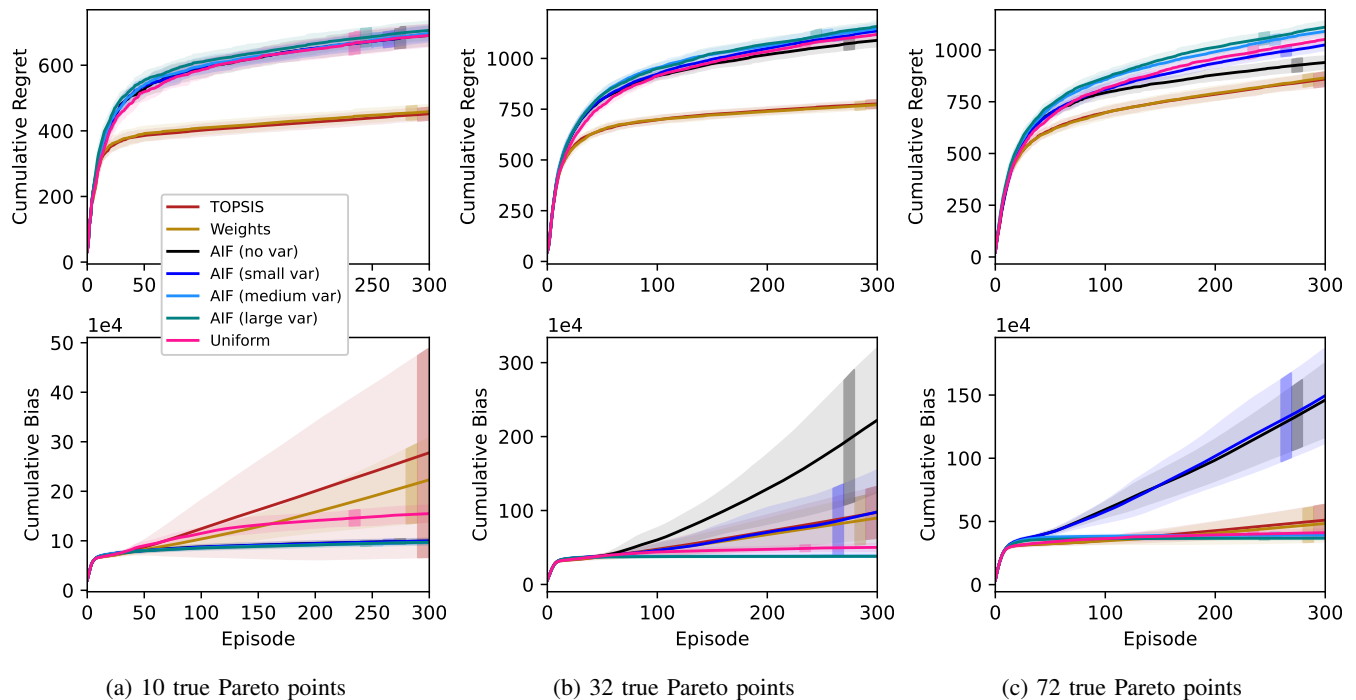


Fig. 6: Specific environment benchmark: Within our framework, we benchmark our active inference (AIF) selection method against other state-of-the-art methods with respect to cumulative Pareto-regret (top) and Pareto-bias (bottom) for three fixed environments with varying numbers of true Pareto points. The mean is represented by the solid line, with 1- σ variance shown by the shaded region. The dark shaded bar aids in distinguishing overlapping variance bands.

in roughly one and a half hours. To express this, the chosen preference distribution is

$$\mathcal{N}(\mu_{\text{pr}}, \Sigma_{\text{pr}}), \quad \mu_{\text{pr}} = (90, 6) \\ \sigma_{1,1} = 140, \quad \sigma_{1,2} = \sigma_{2,1} = -2, \quad \sigma_{2,2} = 70,$$

where $\sigma_{i,j}$ is the i - j element of Σ_{pr} . Soon after deployment (episode 3), the planner only finds candidate plans that go through sand to collect the sample, and then must wash off, seen in Fig. 5a. The robot quickly learns to avoid sand as washing takes very long and collects lots of radiation, incurring a regret of 33.1 units. The robot gradually discovers more optimal behavior seen by episode 25 in Fig. 5b with regret of only 7.89. However, with a bias of 237, the estimate Pareto front does not yet accurately approximate the true Pareto front. In episode 35, seen in Fig. 5c, the rover has slightly improved the learned Pareto front with a bias of 197. To further improve the learned Pareto front, the robot elects to explore a less preferred, information rich trade-off (low time, high radiation). Finally, after 150 episodes, seen in Fig. 5d, the robot has obtained an accurate estimate of the true Pareto front with a bias of only 14.0 and regret of 1.03, and has converged to selecting the most preferred trade-off.

B. Benchmarks

We evaluate the performance of our framework against the state-of-the-art in two benchmark problems.

We benchmarked our active inference (AIF) selection method against other state-of-the-art selection methods: uniform random selection (fair selection), TOPSIS [14] (objective preference), and linear scalarization, i.e. “weights” (subjective preference). Refer to Sec. V for further description of the compared methods.

1) *Fixed Environment Case Study*: Three grid world environments were designed to challenge the learning of different size Pareto fronts. All experiments were run for 300 episodes, with a LCB confidence parameter $\alpha = 0.1$, and $n_s = 300$ Monte Carlo samples, selected via cross validation. See Appendix ?? for more details on the accuracy with respect to the choice of n_s . Fig. 6 shows benchmarks against each environment. Generally, methods that accumulate less bias require learning more of the transition cost model via exploration, and hence accumulate more regret. When learning the small Pareto front, seen in Fig. 6a, AIF performed better than both uniform and the biased methods (weights and TOPSIS) in terms of bias, with comparable regret to uniform. This is due to the advantageous information-gain-based exploration. When the number of Pareto points on the Pareto front increases, seen in Fig. 6b, using *small* and *no*² variance selects for only a portion of the true Pareto front, while the *medium* and *large* variance still envelope the entire Pareto front. With the smaller variance AIF methods, once a good estimate candidate is found, the first term of (7) dominates the selection, and thus abandons the less promising options that are beyond the variance. With an even larger Pareto front seen in Fig. 6c, this effect is exacerbated.

²If the user only wants to specify a desired point in objective-space, they can make the preference distribution covariance very small (*no* variance)

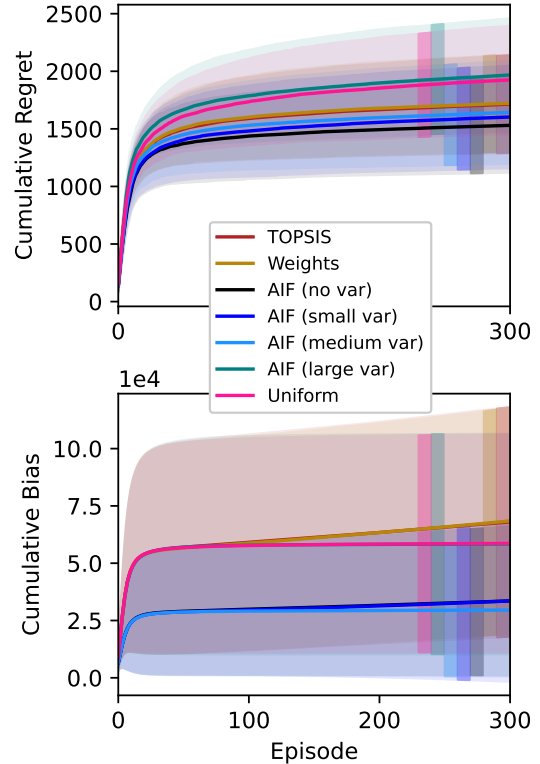


Fig. 7: Randomized environment benchmark: We benchmark our method against other selection methods for cumulative Pareto-regret and Pareto-bias vs. number episodes for randomly generated scenarios of varying complexity. The mean is represented by the solid line with $1\text{-}\sigma$ variance shown by the shaded region.

In this case, the *no* variance method sees a decrease in regret due to the heavy myopic bias after finding a candidate solution that agrees with the preference. Thus, from this experiment, we can observe that AIF methods start becoming very selective once the preferred region (which can be varied by the size of variance) on the Pareto front is discovered.

2) *Random Environment Case Study*: The behavior of our AIF selection method can vary widely across different scenarios depending on the diversity of the Pareto front relative to p_{pr} . Therefore, we illustrate the general behavior of our approach using randomized grid worlds. Each trial, a 20×20 grid world robot model is generated with randomized proposition locations and true cost distribution parameters. The data was analyzed across 750 randomized trials.

Fig. 7 demonstrates the general trade-off between bias and regret. Note that the standard deviation in performance among trials is very large, since each randomly generated environment is diverse. When the variance of p_{pr} is *large*, active inference tends to perform similarly to random selection (Uniform) in terms of both regret and bias. With a moderate variance, active inference tends to have better regret and bias performance than other methods due to the intelligent information theoretic exploration when the estimate plans $\hat{\Pi}^*$ are uninformed. Uniform, AIF (large var), and AIF (medium var) stabilize in cumulative bias, meaning they effectively learn the entire Pareto front. The remaining methods converge to a

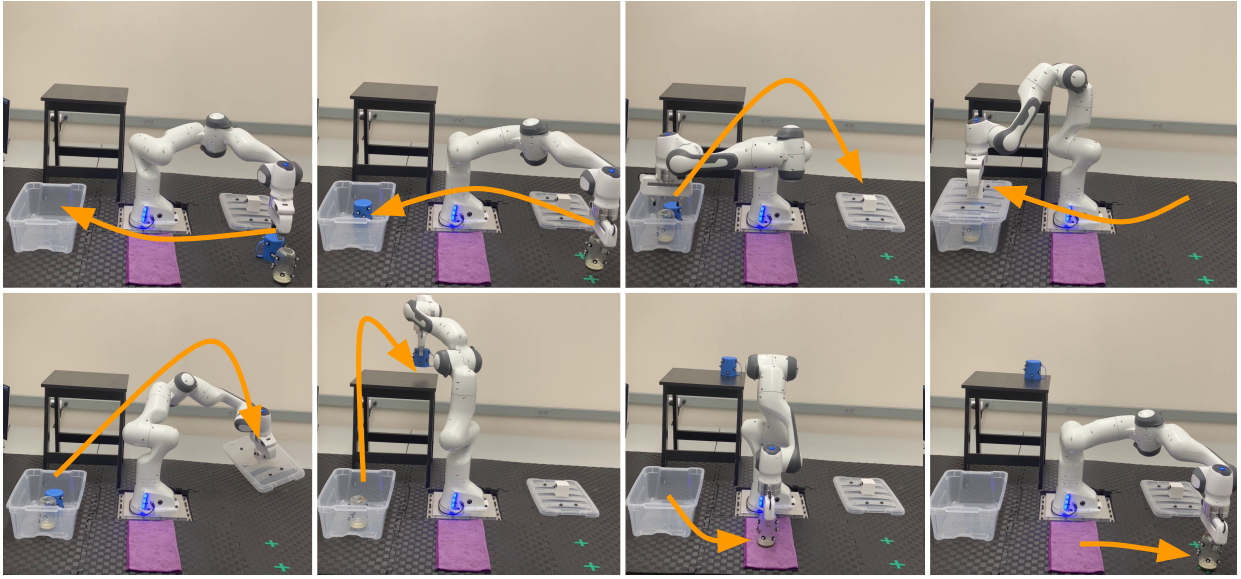


Fig. 8: Hardware dishwashing experiment: A robotic manipulator must load both a durable pitcher and a fragile glass jar into a dishwasher, then must unload each item by either placing it on the drying rack, or manually drying it on the ground. The robot learns to minimize time and risk (holding the jar high above the ground). The user prefers a low-risk, higher-time trade-off. After 500 episodes, the system performs optimally (left to right, top to bottom), and decides to quickly place the pitcher on the rack, and spend time to manually dry the jar. *Video*: <https://youtu.be/tCRJwqeT-f4>.

non-constant slope cumulative bias, indicating that portions of the Pareto front have not been properly learned. From this experiment, we can observe that AIF methods outperform other state-of-the-art methods in terms of both cumulative regrets and biases as long as the size of variance is not too large.

Akin to other expected free energy approaches, the enhanced reasoning power comes at a computational cost compared to other methods [21, 32]. Uniform selection has complexity $O(1)$, weights has $O(|\tilde{\Pi}^*|)$, and TOPSIS has $O(|\tilde{\Pi}^*|^2)$ [44]. We benchmarked the aforementioned random environment case studies with respect to both *planning* time and *selection* time per episode on a computer with AMD Ryzen 7 3800X 3.9 GHz 8-Core Processor. The average planning time across 400 experiments with 100 episodes each is 483 ± 443 ms ($1-\sigma$) with a median planning time of 369ms. The average AIF selection time (across all four preference distribution variances) is 610 ± 1051 ms ($1-\sigma$) with a median selection time of 179ms. The compared selection procedures completed in less than 1ms.

Remark 1. *For larger-scale robotic models, the exact multi-objective graph search can be replaced with approximate multi-objective search, such as A*pex [45], to admit a smaller (approximate) Pareto front in less time. Additionally, a clustering algorithm [46, 47] can further reduce the size of the Pareto front. We expect both adaptations to our framework to greatly enhance computation speed at the cost of Pareto-regret and bias performance, however we leave this to future work.*

C. Hardware Experiment

To demonstrate the efficacy and diversity of application of our framework, we studied a complex dishwashing experiment

using a robotic manipulator described in Fig. 1. The robot must repetitively load and unload the dishwasher with two items, a durable pitcher, and a fragile glass jar. The LTLf specification is

$$\phi = \phi_l \wedge \phi_s,$$

where

$$\begin{aligned} \phi_l = & \diamond(P_{dw} \wedge J_{dw} \wedge lid_{on} \wedge \diamond(P_{rack} \vee (P_{dry} \wedge \diamond P_{floor})) \wedge \\ & \diamond(J_{rack} \vee (J_{dry} \wedge \diamond J_{floor}))), \\ \phi_s = & \square(\neg lid_{aside} \rightarrow \diamond(P_{dw} \wedge J_{dw})), \end{aligned}$$

which can be interpreted as “put the pitcher and the jar in the dishwasher, then put the lid on, then for both the pitcher and jar, place on the drying rack, or dry the item and place on the ground”. The performance of the robot is evaluated by the cumulative execution time (objective 1), as well as cumulative risk (objective 2) measured for a given action primitive,

$$risk(s, a) = \begin{cases} \int_t h_J^2 dt & \text{if holding jar,} \\ 0 & \text{otherwise,} \end{cases} \quad (23)$$

where h_J is the height of the jar above the ground. A sampling based motion planner (PRM* [48]) is used to realize motion action primitives, inducing unknown and random time and risk costs. Note that placing the jar on the drying rack is likely to have high risk, whereas manually drying the jar on the ground will likely take more time. We trained the transition cost model in simulation for 500 episodes, using a preference distribution of $\mu_{pr} = (350, 0.5)$ and $\sigma_{1,1}^2 = 400$, $\sigma_{2,2}^2 = 2$, $\sigma_1 \sigma_2 = 0$, which selects for a high-time, low-risk alternative. The converged behavior is shown in Fig. 8. As can be seen, the robot correctly loads the dishwasher, then unloads the pitcher on the rack, and decides to manually dry the jar, all while not performing any sub-optimal/unnecessary actions. The video of

the execution is included in the supplementary material and visually represents the applicability of our MORL framework to a variety of robotic models with complex tasks.

V. RELATED WORK

A. Multi-Objective Reinforcement Learning

The goal of MORL is to maximize a multi-objective value or utility function [13]. MORL extends single-objective RL methods by selecting among Pareto-optimal candidate actions. One method of selection is casting valuations of candidate actions to a single-objective by scalarizing, in turn requiring a sound and interpretable utility function.

Scalarization of utility functions can be classified as *preference-based*, scalarizing based on an exogenous user’s preferred trade-off [12], or *non-preference-based*, scalarizing without external preferences [11, 14]. Linear scalarization is a standard preference-based method that combines objectives through a relative-importance weighted sum. On the contrary, non-preference-based methods such as TOPSIS [14] and hyper-volume indicators [11] rely on Euclidian distance metrics in objective space. However, in many robotic applications where objectives have different units of measurement (e.g. time in seconds and energy in joules), these methods use the addition of quantities with different units, which is generally mathematically ad-hoc and lacks interpretation. Linear scalarization can potentially combat this issue by interpreting weights as reciprocal measures (e.g. 1/second), at the cost of requiring a user to have a rigorous understanding of how to compare the importance of each objective and properly assign weights. Furthermore, re-interpreting units of measurement (e.g. seconds instead of milliseconds) alters Euclidian distance, which can dramatically alter the selected solution, when no fundamental aspects about the problem have changed. On the contrary, our method avoids this issue by only describing the scalarization through operations on probability density functions.

Many MORL methods do not distinguish between optimizing multi-objective value function and selecting the best action from the Pareto-optimal candidate actions [12, 13]. Instead, these methods scalarize using a monotonic utility function, which guarantees that an optimal (preferred) action with respect to the single-objective utility function must also be Pareto-optimal in objective space. The use of such a utility function effectively optimizes a single objective, which casts a complete order on the reward space and eliminates the need for explicit computation of the Pareto front beforehand. However, when a non-monotonic utility function is used, or the agent is concerned with learning the set of all optimal trade-off solutions [16], the Pareto front must be explicitly computed before *selection*, as seen in many MOMAB approaches. Our framework computes the Pareto front using a multi-objective task-planner.

MOMAB problems are concerned with minimizing *Pareto-Regret* [16], which uses ϵ -*dominance* to measure how “far” the chosen action is from being Pareto-optimal with respect to the true unknown reward distributions. Besides Pareto-regret, the MOMAB performance can also be evaluated by *unfairness*,

which quantifies how evenly Pareto-optimal actions were selected. Unfairness is used to compare MOMAB selection by diversity in selection and learning of all actions on a possibly non-convex Pareto-front. To learn the entire Pareto front, work [16] extends the single-objective Upper Confidence Bound (UCB) algorithm [37] to multiple objectives, then rely on random selection among the Pareto front computed from UCB scores. Furthermore, this work illustrates how random selection of UCB-Pareto-Optimal candidates may still fail to learn non-convex Pareto-fronts using linear scalarization. While random selection effectively learns the entire Pareto front, it cannot account for or converge to a preferred trade-off. To address this shortcoming, our formulation of the high-level active inference decision making agent naturally balances this exploration vs. exploitation trade-off.

B. Active Inference and Expected Free Energy

Active inference is an uncertainty-based sequential decision making scheme that applies the free energy principle (FEP) [49] to the behavioral norms of physical agents [20, 21, 22]. According to the FEP, agents intend to minimize the surprise (in our problem it is denoted as $-\log p(\mathbf{C}|\pi)$) to maintain their homeostasis. However, calculating the surprise directly requires to sum/integrate over all possible states/parameters in their internal generative models. When these quantities are continuous, depending on the prior and/or the likelihood function, the integral does not have a closed-form solution. Thus, some approximation methods are required to calculate the surprise.

One of the approaches is to employ variational Bayesian inference in statistical machine learning [41]. The core idea of this methodology is to introduce a proposal/surrogate distribution, which is *easily* modeled, and optimize the functional of this distribution to make it resemble a true intractable distribution. More specifically, in our problem, the following free-energy (FE, i.e. negative evidence lower bound) is minimized such that the Kullback-Leibler Divergence (KLD) between the proposal distribution $q(s, \tilde{\theta}|\pi)$ and the true distribution $p(s, \theta|\mathbf{C}, \pi)$ approaches zero (note that here we used the simplified notations seen in Appendices ?? and ??),

$$\begin{aligned} FE &= \int_{s, \tilde{\theta}} q(s, \tilde{\theta}|\pi) \log \frac{q(s, \tilde{\theta}|\pi)}{p(s, \tilde{\theta}, \mathbf{C}|\pi)} d\tilde{\theta} ds \\ &= \int_{s, \tilde{\theta}} q(s, \tilde{\theta}|\pi) \log \frac{q(s, \tilde{\theta}|\pi)}{p(s, \tilde{\theta}|\mathbf{C}, \pi)} d\tilde{\theta} ds - \log p(\mathbf{C}|\pi) \\ &= D_{KL} \left(q(s, \tilde{\theta}|\pi) || p(s, \theta|\mathbf{C}, \pi) \right) - \log p(\mathbf{C}|\pi) \\ &\geq \text{Surprise}(\pi). \end{aligned} \tag{24}$$

As can be seen from this equation, since KLD is greater or equal to zero, FE is the upper bound of the surprise, i.e. minimizing FE leads to minimize the surprise.

Computation of FE is performed given an observed outcome observation \mathbf{C} , which is not known until the plan is executed. Hence, in active inference, the expected free energy (EFE) is instead minimized, and similar to FE, if the proposal and the true hidden posterior distributions match, the EFE term

reduces to the following,

$$EFE \rightarrow -\mathbb{E}_{q(\mathbf{C}|\pi)}[\log p_{\text{Pr}}(\mathbf{C})], \quad (25)$$

which is “expected” surprise. In summary, FE allows to make the surprise minimization tractable for continuous state/parameter systems, and EFE allows for the minimization FE without knowing the received observation *a priori*.

Additionally, as explained in (6), the EFE comprises of an utility term governed by a prior agent’s desired observation distribution, and an information gain term that evaluates how much a candidate action would reduce the uncertainty of hidden states/parameters. Hence, by minimizing EFE, the agents can naturally balance between two modes of behavior that are essential for autonomous decision making under uncertainty; *exploitation* (i.e. focusing on the current best action) and *exploration* (i.e. trying less executed actions). Furthermore, the degree of preference for either mode can be easily adjusted by varying the prior preference distribution. For instance, using a prior preference with a large variance will cause the agents to provide more priority to exploration, and vice versa. These characteristics are particularly useful when transparent agent behavior is desired [50]. Using active inference for *selection* marries the benefits of *preferred* versus *fair* selection described in Sec. V-A through localized information-theoretic exploration around the prior preference distribution. Additionally, specifying a preference distribution over desired trade-offs may be particularly intuitive, explainable, and even statistically informed with past data. Throughout this work, we study the benefits of using EFE in active inference for Pareto point selection in MORL with two statistical techniques when an intractable convolved NIW is used to represent the hidden parameters.

C. Task and Motion Planning

Task and Motion Planning (TAMP) is concerned with integrating high-level task planning over abstracted robotic action-primitives with a low-level motion planner responsible for computing feasible plans for each action-primitive in continuous space [51]. TAMP methods are primarily concerned with assessing both task and motion feasibility, however there do exist works on informing high-level costs based on low-level feasibility [28] as well as directly optimizing low-level cost objectives [52]. Our work tackles a slightly different problem than TAMP, differing in three main aspects: i) cost representations do not need to pertain to motion, ii) robotic actions need not be motion based, and iii) cost simulations/models may not be available at planning time. Nonetheless, our framework may be used to solve TAMP problems if low-level motion feedback at planning time is not a primary concern.

VI. CONCLUSION AND FUTURE WORK

In this work we study a systematic approach to learning Pareto-optimal behavior on an unknown multi-objective stochastic cost model. In particular we examine an active inference-based approach to multi-objective decision making and we examine the resulting behavior through robotic simulation and hardware experiments, and numerical benchmarks

that compare against other state-of-the-art selection methods. Our approach marries the benefits of a user-provided preferred optimal trade-off with exploration required to learn portions of the entire Pareto front. We introduce a new Pareto-bias metric that, coupled with traditional Pareto-regret, elucidates the trade-off between accurately learning the diverse Pareto front versus quickly learning a single optimal trade-off. Notably, the balance between the aforementioned behaviors can be controlled via the user’s preference.

This work gives rise to a few future directions of research such as extension to stochastic transition models. Additionally, integrating cost-bounds for certain objectives by embedding a pruning procedure [10] into the multi-objective graph search algorithm is an interesting extension. Since the costs are not known *a priori*, we expect the Pareto cost *upper* confidence bound to be an effective choice of pruning constraint. We are also interested in generalizing the cost distribution to a Gaussian mixture to account for multi-modality. Other future directions may include convergence analysis and finite time regret analysis, for example, adapting the Upper Credible Limit algorithm [34] for Gaussian multi-armed bandits to the multi-objective setting.

REFERENCES

- [1] Yang Gao and Steve Chien. Review on space robotics: toward top-level science through space exploration. *Science Robotics*, 2(7), 2017.
- [2] Jaeseok Kim, Anand Kumar Mishra, Raffaele Limosani, Marco Scafuro, Nino Cauli, Jose Santos-Victor, Barbara Mazzolai, and Filippo Cavallo. Control strategies for cleaning robots in domestic applications: A comprehensive review. *International Journal of Advanced Robotic Systems*, 16(4):1729881419857432, 2019.
- [3] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 1996.
- [4] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.
- [5] Bharat Singh, Rajesh Kumar, and Vinay Pratap Singh. Reinforcement learning in robotic applications: a comprehensive survey. *Artificial Intelligence Review*, 55(2):945–990, 2022.
- [6] Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021.
- [7] Hadas Kress-Gazit, Morteza Lahijanian, and Vasumathi Raman. Synthesis for robots: Guarantees and feedback for robot behavior. *Review of Control, Robotics, and Autonomous Systems*, 1:211–236, May 2018.
- [8] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking (Representation and Mind Series)*. The MIT Press, 2008.
- [9] Giuseppe De Giacomo and Moshe Y Vardi. Linear temporal logic and linear dynamic logic on finite traces. In *IJCAI’13 Proceedings of the Twenty-Third international*

- joint conference on Artificial Intelligence, pages 854–860. Association for Computing Machinery, 2013.
- [10] Peter Amorese and Morteza Lahijanian. Optimal cost-preference trade-off planning with multiple temporal tasks. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2071–2077, 2023.
- [11] Weijia Wang and Michèle Sebag. Multi-objective Monte-Carlo tree search. In Steven C. H. Hoi and Wray Buntine, editors, *Proceedings of the Asian Conference on Machine Learning*, volume 25 of *Proceedings of Machine Learning Research*, pages 507–522, Singapore Management University, Singapore, 04–06 Nov 2012. PMLR.
- [12] Axel Abels, Diederik M. Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic weights in multi-objective deep reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 11–20. PMLR, 2019.
- [13] Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Raymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irisappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1), apr 2022.
- [14] Mohammad Mirzanejad, Morteza Ebrahimi, Peter Vamplew, and Hadi Veisi. An online scalarization multi-objective reinforcement learning algorithm: Topsis q-learning. *The Knowledge Engineering Review*, 37:e7, 2022.
- [15] Slim Bechikh, Marouane Kessentini, Lamjed Ben Said, and Khaled Ghédira. Preference incorporation in evolutionary multiobjective optimization: A survey of the state-of-the-art. In *Advances in computers*, volume 98, pages 141–207. Elsevier, 2015.
- [16] Madalina M. Drugan and Ann Nowe. Designing multi-objective multi-armed bandits algorithms: A study. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2013.
- [17] Róbert Busa-Fekete, Balázs Szörényi, Paul Weng, and Shie Mannor. Multi-objective bandits: Optimizing the generalized gini index. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17*, page 625–634. JMLR.org, 2017.
- [18] Eralp Turgay, Doruk Öner, and Cem Tekin. Multi-objective contextual bandit problem with similarity information. In Amos J. Storkey and Fernando Pérez-Cruz, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, volume 84 of *Proceedings of Machine Learning Research*, pages 1673–1681. PMLR, 2018.
- [19] Shiyin Lu, Guanghui Wang, Yao Hu, and Lijun Zhang. Multi-objective generalized linear bandits. *IJCAI'19*, page 3080–3086. AAAI Press, 2019.
- [20] Karl Friston, Francesco Rigoli, Dimitri Ognibene, Christoph Mathys, Thomas FitzGerald, and Giovanni Pezzulo. Active inference and epistemic value. *Cognitive neuroscience*, 02 2015.
- [21] Raphael Kaplan and Karl J. Friston. Planning and navigation as active inference. *Biol. Cybern.*, 112(4):323–343, aug 2018.
- [22] Karl J Friston, Maxwell JD Ramstead, Alex B Kiefer, Alexander Tschantz, Christopher L Buckley, Mahault Albarracin, Riddhi J Pitliya, Conor Heins, Brennan Klein, Beren Millidge, Dalton AR Sakthivadivel, Toby St Clere Smithe, Magnus Koudahl, Safae Essafi Tremblay, Capm Petersen, Kaiser Fung, Jason G Fox, Steven Swanson, Dan Mapes, and Gabriel René. Designing ecosystems of intelligence from first principles. *Collective Intelligence*, 3(1):26339137231222481, 2024.
- [23] Shohei Wakayama and Nisar Ahmed. Active inference for autonomous decision-making with contextual multi-armed bandits. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7916–7922, 2023.
- [24] Marius Kloetzer and Calin Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 53(1):287–297, 2008.
- [25] Georgios E Fainekos, Hadas Kress-Gazit, and George J Pappas. Temporal logic motion planning for mobile robots. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2020–2025. IEEE, 2005.
- [26] Shuo Yang, Xiang Yin, Shaoyuan Li, and Majid Zamani. Secure-by-construction optimal path planning for linear temporal logic tasks. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 4460–4466. IEEE, 2020.
- [27] Georgios E Fainekos, Hadas Kress-Gazit, and George J Pappas. Hybrid controllers for path planning: A temporal logic approach. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 4885–4890. IEEE, 2005.
- [28] Keliang He, Morteza Lahijanian, Lydia E Kavraki, and Moshe Y Vardi. Towards manipulation planning with temporal logic specifications. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 346–352. IEEE, 2015.
- [29] Keliang He, Morteza Lahijanian, E Kavraki, Lydia, and Y Vardi, Moshe. Automated abstraction of manipulation domains for cost-based reactive synthesis. *IEEE Robotics and Automation Letters*, 4(2):285–292, Apr. 2019.
- [30] Pablo Lanillos, Cristian Meo, Corrado Pezzato, Ajith Anil Meera, Mohamed Baioumy, Wataru Ohata, Alexander Tschantz, Beren Millidge, Martijn Wisse, Christopher L. Buckley, and Jun Tani. Active inference in robotics and artificial agents: Survey and challenges. *CoRR*, abs/2112.01871, 2021.

- [31] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006.
- [32] Ryan Smith, Karl J. Friston, and Christopher J. Whyte. A step-by-step tutorial on active inference and its application to empirical data. *Journal of Mathematical Psychology*, 107:102632, 2022.
- [33] Kevin P. Murphy. Conjugate bayesian analysis of the gaussian distribution. Technical report, University of British Columbia, 2007.
- [34] Paul B Reverdy, Vaibhav Srivastava, and Naomi Ehrich Leonard. Modeling human decision making in generalized gaussian multiarmed bandits. *Proceedings of the IEEE*, 102(4):544–571, 2014.
- [35] Yen-Chi Chen. A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology*, 1(1):161–187, 2017.
- [36] Lawrence Mandow and José Luis Pérez De La Cruz. Multiobjective a* search with consistent heuristics. *Journal of the ACM (JACM)*, 57(5):1–25, 2008.
- [37] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2–3):235–256, may 2002.
- [38] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.
- [39] Horia Mania, Stephen Tu, and Benjamin Recht. Certainty equivalence is efficient for linear quadratic control. In *Advances in Neural Information Processing Systems*, volume 32, pages 10154–10164, 2019.
- [40] Sang Gyu Kwak and Jong Hae Kim. Central limit theorem: the cornerstone of modern statistics. *Korean journal of anesthesiology*, 70(2):144–156, 2017.
- [41] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.
- [42] William L Dunn and J Kenneth Shultis. *Exploring monte carlo methods*. Elsevier, 2022.
- [43] Kenneth A Farley, Kenneth H Williford, Kathryn M Stack, Rohit Bhartia, Al Chen, Manuel de la Torre, Kevin Hand, Yulia Goreva, Christopher DK Herd, Ricardo Hueso, et al. Mars 2020 mission overview. *Space Science Reviews*, 216:1–41, 2020.
- [44] Hamdani and Retantyo Wardoyo. The complexity calculation for group decision making using topsis algorithm. In *AIP conference proceedings*, volume 1755, page 070007. AIP Publishing LLC, 2016.
- [45] Han Zhang, Oren Salzman, TK Satish Kumar, Ariel Felner, Carlos Hernández Ulloa, and Sven Koenig. A* pex: Efficient approximate multi-objective search on graphs. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 32, pages 394–403, 2022.
- [46] Enrico Zio and Roberta Bazzo. A clustering procedure for reducing the number of representative solutions in the pareto front of multiobjective optimization problems. *European Journal of Operational Research*, 210(3):624–634, 2011.
- [47] Lilian Astrid Bejarano, Helbert Eduardo Espitia, and Carlos Enrique Montenegro. Clustering analysis for the pareto optimal front in multi-objective optimization. *Computation*, 10(3):37, 2022.
- [48] SM LaValle. *Planning Algorithms*, volume 2. 2006.
- [49] K. Friston. The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, 11(2):127–138, 2010.
- [50] Shohei Wakayama and Nisar Ahmed. Observation-augmented contextual multi-armed bandits for robotic search and exploration. *IEEE Robotics and Automation Letters*, 2024.
- [51] Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4(1):265–293, 2021.
- [52] Marco Faroni, Alessandro Umbrico, Manuel Beschi, Andrea Orlandini, Amedeo Cesta, and Nicola Pedrocchi. Optimal task and motion planning and execution for multiagent systems in dynamic environments. *IEEE Transactions on Cybernetics*, 54(6):3366–3377, 2023.