Hierarchical Probabilistic Intention Inference in Partially Observable Environments

***Abstract*—nferring latent intentions from sequential behavior is challenging when a decision-maker's objectives are not directly observable and its behavior may be sub-optimal or dynamically changing, particularly in partially observable environments with multiple high-level subtasks. Existing approaches, such as trajectory prediction and supervised intent recognition, are often domain-specific and may not explicitly capture preferences, priorities, and irregular decision patterns. This paper presents a probabilistic intent inference framework based on a hierarchical behavior model for bounded-rational decision-makers within a partially observable Markov decision process (POMDP).nferring latent intentions from sequential behavior is challenging when a decision-maker's objectives are not directly observable and its behavior may be sub-optimal or dynamically changing, particularly in partially observable environments with multiple high-level subtasks. Existing approaches, such as trajectory prediction and supervised intent recognition, are often domain-specific and may not explicitly capture preferences, priorities, and irregular decision patterns. This paper presents a probabilistic intent inference framework based on a hierarchical behavior model for bounded-rational decision-makers within a partially observable Markov decision process (POMDP).I The model captures decision-making at both the action and subtask levels, using three interpretable parameters to characterize low-level rationality, high-level subtask selection, and persistence in task execution. We derive a Bayesian recursion for the intent posterior from noisy observations that is exact under the stated model assumptions. A Monte Carlo predictive prior supports online subtask-level prediction. Analytical results characterize the model's performance across different rationality settings, while extensive simulations in 2D and 3D environments demonstrate improved accuracy, adaptability, and robustness compared to representative probabilistic and sequence-model baselines. We further evaluate robustness on heterogeneous sub-optimal trajectories generated by large language model (LLM)-based proxy agents.**

***Index Terms*—Intention inference, bounded-rational behavior modeling, partially observable Markov decision process (POMDP), reinforcement learning.**

I. Introduction

In recent years, the integration of artificial intelligence (AI) agents into collaborative environments has rapidly expanded across domains such as healthcare, robotics, and autonomous systems. In many of these settings, a critical requirement is the ability of AI agents not only to process observed actions and trajectories of their collaborators, but also to infer the latent intentions underlying observed behavior. This capability becomes particularly important in complex and time-sensitive tasks, where AI agents must infer likely objectives from behavior despite partial observability and limited communication. Human decision-makers in human–AI collaborative settings provide a primary motivating example of this challenge: their internal objectives are not directly observable, their behavior may be bounded-rational rather than fully optimal, and their priorities may shift during task execution. Accordingly, we adopt a simplified yet structured behavioral model that enables tractable, interpretable latent-intent inference, drawing on established cognitive science foundations including resource-rational decision-making [1], [2], cognitive stability and task switching [3], and hierarchical planning under Theory of Mind [4]–[6].

For the human–AI collaborative settings described above, several methods have been developed to predict underlying human intent over time from sequential behavioral observations [7]–[13]. Two main approaches are supervised learning algorithms [14]–[23] and probabilistic methods [24]–[29]. Supervised learning methods utilize various classification or prediction models, such as neural networks and Gaussian process regression, to estimate human intent [30], [31]. These methods require large amounts of labeled human data for training, which limits their applicability in data-scarce domains. Additionally, they are largely deterministic and lack the ability to quantify prediction confidence or expected uncertainty.

Probabilistic methods, on the other hand, aim to recursively compute the posterior probability of human intent as new data is observed. They do not rely on large datasets, and their probabilistic nature enables the assessment of inference uncertainty. For instance, the recursive Bayesian approaches introduced in [29], [30], [32], [33] estimate intention probabilities using noisy observations. These methods often approximate observation likelihoods based on the Euclidean distance between observations and subtask locations, independent of past trajectories, but do not account for human decision-making or temporal changes in behavior. Other probabilistic approaches proposed in [28], [34] model low-level behavior using a reinforcement learning agent. This approach requires state-action pairs, which are often unavailable due to limited access to actions and noise in observational data.

In addition to general human–robot interaction, driver intention recognition in autonomous driving offers a salient example of intent inference under uncertainty. Prior work in this area uses deep learning models to infer latent driver goals from steering, acceleration, and gaze [35]–[42]. These approaches improve coordination and safety in dynamic environments. However, they are often limited to single-shot or deterministic reasoning and do not generalize to multi-subtask collaborative settings.

Recent work has also explored the use of large language models (LLMs) for intent recognition, particularly in language-based domains such as dialog systems, command interpretation, and task-oriented assistance [43], [44]. These methods leverage LLMs' ability to reason over unstructured inputs and generalize across tasks in a zero-shot or few-shot manner. While effective at extracting high-level intent from

textual cues, they typically lack access to structured environmental states or physical trajectories and are not designed for stepwise inference in dynamic, noisy settings. Meanwhile, recent advances in modeling decision behavior have focused on techniques such as inverse reinforcement learning and imitation learning [45]–[49], which aim to deduce rewards or policies from observed behaviors. Despite their strengths, these methods often overlook the hierarchical nature of decision-making in collaborative settings.

Supervised sequence models for intent recognition, including LSTM and attention-based architectures, provide expressive trajectory encoders when labeled trajectories are available. The proposed framework addresses a complementary setting in which a structured POMDP model is specified and the observer receives partial, noisy observations. In this setting, the framework maintains a recursive posterior over latent intent and represents subtask switching and persistence through interpretable behavioral parameters. Section VI reports representative probabilistic baselines, recurrent sequence baselines, and a zero-shot LLM observer as an auxiliary reference. A separately trained Transformer comparison would require a dedicated architecture-selection and hyperparameter-tuning study, and empirical claims are limited to the evaluated baselines.

To address these limitations, this paper introduces a hierarchical probabilistic framework for inferring intent over a set of subtasks from partial and noisy observations using a partially observable Markov decision process (POMDP). The proposed model captures decision structure at three levels: (1) action-level rationality, modeled using ϵ-greedy behavior; (2) high-level subtask rationality, reflecting preferences and priorities in subtask selection; and (3) persistence, characterizing the tendency to remain committed to the current subtask versus switching over time. Based on this hierarchical model, we derive a Bayesian recursive posterior over latent intent that is exact under the stated assumptions on the candidate subtask set, completion feedback, behavioral model, switching model, and POMDP transition and observation models. The next-subtask predictive prior is estimated by Monte Carlo rollouts. We also show how the behavioral parameters can be inferred jointly with latent intent when they are unknown a priori.

Fig. 1 illustrates these three behavioral factors through example trajectories. The left plot visualizes action-level rationality, where the green path represents sub-optimal behavior in reaching a subtask, while the black path represents a more rational trajectory. The center plot highlights differences in subtask selection, where the black paths correspond to more rational high-level choices and the red dashed paths indicate less rational subtask selection. The right plot visualizes persistence, where the black path represents commitment to the current subtask and the red dashed paths indicate possible switching between subtasks during execution. Such switching may arise from uncertainty, imperfect planning, changing priorities, or other sources of bounded-rational behavioral variability.

The main contributions of this paper include:

- Deriving the posterior probability of latent intent from noisy behavioral trajectories without requiring direct access to the underlying actions.
- Incorporating interpretable preference and priority structures into intention inference, enabling applicability across a range of environments without relying on hand-crafted distance-based likelihoods or domain-specific task representations.
- Supporting the prediction of future subtasks and likely action tendencies, thereby capturing both abrupt and gradual changes in behavior over time.

Our analytical results characterize the performance of the proposed method across different rationality regimes and under direct state observability. The numerical experiments demonstrate strong performance compared with several representative intention recognition methods in both two- and three-dimensional environments. Together, these results highlight the method's ability to accurately infer latent intent and anticipate future behavior under varying levels of rationality and uncertainty, supporting its robustness and applicability in complex collaborative settings.

The remainder of this paper is organized as follows: Section II outlines the fundamental principles of the proposed intention inference approach. Section III provides a detailed description of the behavioral model employed in the methodology. In Section IV, we discuss the recursive computation of the intention posterior. Section V analyzes the influence of key behavioral parameters on the performance of the proposed method. Section VI presents the results of numerical experiments, and finally, Section VII offers concluding remarks and discusses potential directions for future research.

II. Proposed Intention Inference Approach

A. Partially Observable Markov Decision Process

A partially observable Markov decision process (POMDP) [11], [12], [50], [51] is a useful model for representing decision-making under uncertainty in stochastic environments. In this paper, we use a POMDP to represent bounded-rational decision behavior from partial observations, defined by the tuple $\langle \mathcal{S}, \mathcal{Z}, \mathcal{A}, \mathcal{O}, \mathcal{P}, R \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, and $\mathcal{P}$ is the state transition probability function, with $\mathcal{P}(\mathbf{s}, \mathbf{a}, \mathbf{s}') = P(\mathbf{s}' \mid \mathbf{s}, \mathbf{a})$ representing the probability that the decision-maker moves to state $\mathbf{s}'$ when action $\mathbf{a}$ is taken in state $\mathbf{s}$. $\mathcal{Z}$ denotes the observation space, and $\mathcal{O}$ is the observation model, where $\mathcal{O}(\mathbf{z}, \mathbf{s}) = P(\mathbf{z} \mid \mathbf{s})$ indicates the probability of observing $\mathbf{z}$ when the true state is $\mathbf{s}$. R is a bounded reward function with real-valued outcomes, where $R(\mathbf{s}, \mathbf{a}, \mathbf{s}')$ encodes the reward obtained when action $\mathbf{a}$ is taken in state $\mathbf{s}$ and the decision-maker transitions to state $\mathbf{s}'$.

B. Subtasks as High-Level Intentions

We consider a scenario where a bounded-rational decision-maker is executing a high-level task in a complex environment, which can be decomposed into N pre-specified subtasks, represented as $\{T^1, \ldots, T^N\}$. In an idealized setting, the decision-maker would complete the subtasks in order of their relative importance or priority. However, these priorities vary across individuals, are often known only to this decision-maker, and may remain hidden from external observers (e.g.,

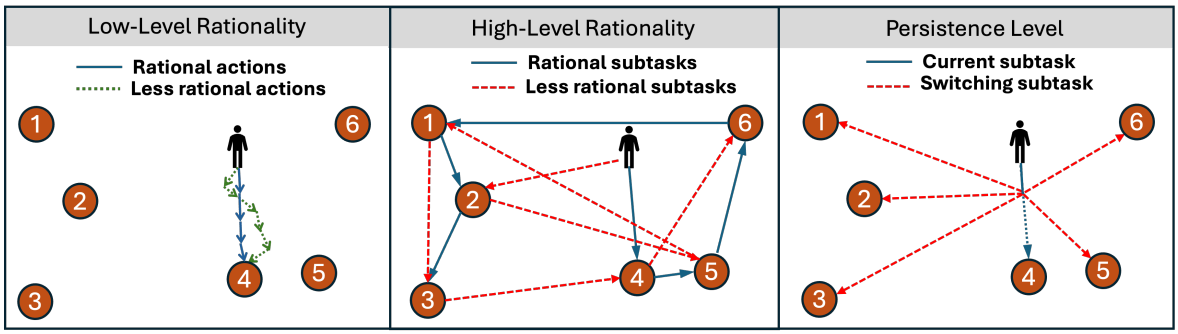


Figure 1: An illustration of the three key behavioral factors in our model: **Left (Low-Level Rationality):** The green path illustrates sub-optimal actions, while the black path represents a more rational trajectory toward subtask completion. **Center (High-Level Rationality):** Black paths demonstrate more rational decision-making at the subtask level, while red dashed paths indicate less rational choices in subtask selection. **Right (Persistence):** The black path shows persistence in completing the current subtask, whereas red dashed paths represent possible switching between subtasks during operation.

AI agents). Moreover, priorities can fluctuate over time as conditions change. These factors make inferring latent intent from behavioral data both challenging and crucial for effective collaboration.

Let $p_0 = [p(T_0 = T^1), \ldots, p(T_0 = T^N)]$ represent the initial intent probability vector, where $p_0(i)$ indicates the probability of initially aiming to perform subtask T^i before observing any new data, and $\sum_{i=1}^{N} p_0(i) = 1$. We assume that $\{\mathcal{G}^1, \ldots, \mathcal{G}^N\}$ is the set of terminal states corresponding to the completion of each subtask, meaning that if the decision-maker reaches $\mathbf{s} = \mathcal{G}^i$, the subtask T^i is completed. Moreover, we assume that only a sequence of noisy observations is available, without direct access to the underlying action sequence or intent labels. Let $\mathbf{z}_{1:k} = (\mathbf{z}_1, \ldots, \mathbf{z}_k)$ represent the noisy observation data available from time step 1 to k. The main objective is to compute the intent posterior at time step k from the available observation history as $p_k = [p(T_k = T^1 \mid \mathbf{z}_{1:k}), ..., p(T_k = T^N \mid \mathbf{z}_{1:k})]^T$, where p_k represents the probability distribution over the subtasks that may be active at time step k, with $\sum_{i=1}^{N} p_k(i) = 1$. A peaked posterior indicates high-confidence inference, while a flat posterior reflects intent ambiguity.

The present formulation assumes a fixed candidate subtask set and access to subtask-completion feedback when a subtask terminates. Settings with dynamically emerging subtasks or delayed completion signals require augmenting the latent state and are discussed as future extensions in Section VII.

Most existing intention recognition approaches [7], [9] depend on the availability of action sequences, as is common in imitation learning and inverse reinforcement learning techniques for reward or policy learning [6], [45], [47], [48], [52]. However, when action sequences are unavailable, it becomes necessary to reason about the unobserved actions and their connection to intent. Our proposed approach to computing the posterior over subtask intentions is described below.

C. Recursive Posterior of Intentions

Let p_k represent the posterior distribution over latent intent at time step k, given the available observation history $\mathbf{z}_{1:k}$. When a new observation $\mathbf{z}_{k+1}$ is obtained at time step $k+1$, the intent posterior must be updated recursively. This update can be expressed using Bayes' rule as:

$$\begin{aligned} p_{k+1}(j) &= P(T_{k+1} = T^j \mid \mathbf{z}_{1:k+1}) \\ &= \frac{P(\mathbf{z}_{k+1}, T_{k+1} = T^j \mid \mathbf{z}_{1:k})}{\sum_{r=1}^{N} P(\mathbf{z}_{k+1}, T_{k+1} = T^r \mid \mathbf{z}_{1:k})} \\ &= \frac{P(\mathbf{z}_{k+1} \mid \mathbf{z}_{1:k}, T_{k+1} = T^j) P(T_{k+1} = T^j \mid \mathbf{z}_{1:k})}{\sum_{r=1}^{N} P(\mathbf{z}_{k+1} \mid \mathbf{z}_{1:k}, T_{k+1} = T^r) P(T_{k+1} = T^r \mid \mathbf{z}_{1:k})} \end{aligned} \tag{1}$$

for $j = 1, ..., N$. The denominator in Eq. (1) serves as a normalizing factor, ensuring that the posterior probabilities sum to 1. The numerator consists of two terms, with the second term expanded as:

$$\begin{aligned} P(T_{k+1} = T^j \mid \mathbf{z}_{1:k}) &= \sum_{l=1}^{N} P(T_{k+1} = T^j, T_k = T^l \mid \mathbf{z}_{1:k}) \\ &= \sum_{l=1}^{N} P(T_{k+1} = T^j \mid T_k = T^l, \mathbf{z}_{1:k}) P(T_k = T^l \mid \mathbf{z}_{1:k}) \\ &= \sum_{l=1}^{N} \underbrace{P(T_{k+1} = T^j \mid T_k = T^l)}_{\text{Switching Probability: } \beta(j|l)} \; p_k(l) \end{aligned} \tag{2}$$

where $p_k(l)$ is the intent posterior at time step k, and $\beta(j \mid l)$ represents the probability that the decision-maker switches from subtask T^l to subtask T^j, which is assumed to be time- and history-independent. These assumptions enable representing the switching probability in Eq. (2) through a single parameter as:

$$\beta(j \mid l) = \begin{cases} c, & j = l, \\ (1-c)/|\mathcal{J}(l)|, & j \in \mathcal{J}(l), \\ 0, & \text{otherwise}, \end{cases}$$

where $\mathcal{J}(l)$ denotes the set of admissible subtasks other than T^l to which switching is allowed, and $0 \le c \le 1$ is the *persistence parameter*, representing the probability of not switching subtasks at any given time. A value of $c = 1$ represents persistent completion of a selected subtask, whereas a smaller value represents more frequent subtask changes. This reflects the well-documented cognitive cost of task switching [3], [53].

Next, the first term in the numerator of Eq. (1) can be expanded as:

$$
\begin{aligned}
&P(\mathbf{z}_{k+1} \mid \mathbf{z}_{1:k}, T_{k+1} = T^j) \\
&= \sum_{\mathbf{a}\in\mathcal{A}} \sum_{\mathbf{s}^n\in\mathcal{S}} \sum_{l=1}^{N} P(\mathbf{z}_{k+1}, \mathbf{a}_k = \mathbf{a}, \mathbf{s}_k = \mathbf{s}^n, T_k = T^l \,|\, \mathbf{z}_{1:k}, T_{k+1} = T^j) \\
&= \sum_{\mathbf{a}\in\mathcal{A}} \sum_{\mathbf{s}^n\in\mathcal{S}} \sum_{l=1}^{N} \underbrace{P(\mathbf{z}_{k+1} \,|\, \mathbf{a}_k = \mathbf{a}, \mathbf{s}_k = \mathbf{s}^n, \mathbf{z}_{1:k}, T_k = T^l, T_{k+1} = T^j)}_{\text{Transition Likelihood}} \\
&\qquad \underbrace{P(\mathbf{a}_k = \mathbf{a}, \mathbf{s}_k = \mathbf{s}^n, T_k = T^l \mid \mathbf{z}_{1:k}, T_{k+1} = T^j)}_{\text{Action Likelihood}},
\end{aligned}
\tag{3}
$$

for $j = 1, ..., N$; where, in the second line, the action, state, and intention from the previous time step are augmented to form a joint probability distribution. The transition likelihood measures the probability of the next observation given the past state and action, past observations, and both the current and past intentions. Meanwhile, the action likelihood represents the joint probability distribution of the previous action and state (both of which are unobserved) given the history of observations and the current intention. In the following paragraphs, we provide a simplification of the transition and action likelihoods.

Transition Likelihood Computation: The transition likelihood in Eq. (3) can be further simplified as:

$$
\begin{aligned}
&P(\mathbf{z}_{k+1} \,|\, \mathbf{a}_k = \mathbf{a}, \mathbf{s}_k = \mathbf{s}^n, \mathbf{z}_{1:k}, T_k = T^l, T_{k+1} = T^j) \\
&= \sum_{\mathbf{s}^i\in\mathcal{S}} P(\mathbf{z}_{k+1}, \mathbf{s}_{k+1} = \mathbf{s}^i \,|\, \mathbf{a}_k = \mathbf{a}, \mathbf{s}_k = \mathbf{s}^n, \mathbf{z}_{1:k}, T_k = T^l, T_{k+1} = T^j) \\
&= \sum_{\mathbf{s}^i\in\mathcal{S}} P(\mathbf{z}_{k+1} \mid \mathbf{s}_{k+1} = \mathbf{s}^i)\; P(\mathbf{s}_{k+1} = \mathbf{s}^i \mid \mathbf{a}_k = \mathbf{a}, \mathbf{s}_k = \mathbf{s}^n) \\
&= \sum_{\mathbf{s}^i\in\mathcal{S}} \mathcal{O}(\mathbf{z}_{k+1}, \mathbf{s}^i)\; \mathcal{P}(\mathbf{s}^n, \mathbf{a}, \mathbf{s}^i)
\end{aligned}
\tag{4}
$$

where $\mathcal{O}(\mathbf{z}_{k+1}, \mathbf{s}^i)$ and $\mathcal{P}(\mathbf{s}^n, \mathbf{a}, \mathbf{s}^i)$ are the POMDP observation and transition models. It can be seen from Eq. (4) that the transition likelihood is independent of the latent intention.

Action Likelihood Computation: The action likelihood term in Eq. (3) can be further simplified as:

$$
\begin{aligned}
&P(\mathbf{a}_k = \mathbf{a}, \mathbf{s}_k = \mathbf{s}^n, T_k = T^l \mid \mathbf{z}_{1:k}, T_{k+1} = T^j) \\
&= P(\mathbf{a}_k = \mathbf{a} \mid \mathbf{s}_k = \mathbf{s}^n, \mathbf{z}_{1:k}, T_k = T^l, T_{k+1} = T^j) \\
&\qquad\qquad P(\mathbf{s}_k = \mathbf{s}^n, T_k = T^l \mid \mathbf{z}_{1:k}, T_{k+1} = T^j) \\
&= \underbrace{P(\mathbf{a}_k = \mathbf{a} \mid \mathbf{s}_k = \mathbf{s}^n, \mathbf{z}_{1:k}, T_k = T^l)}_{\text{Behavioral Model}} \\
&\underbrace{P(\mathbf{s}_k = \mathbf{s}^n \mid \mathbf{z}_{1:k}, T_k = T^l)}_{\text{Intent-Specific State Distribution}}\; P(T_k = T^l \mid \mathbf{z}_{1:k}, T_{k+1} = T^j)
\end{aligned}
\tag{5}
$$

where

$$
\begin{aligned}
&P(T_k = T^l \mid \mathbf{z}_{1:k}, T_{k+1} = T^j) \\
&= \frac{P(T_{k+1} = T^j \mid T_k = T^l, \mathbf{z}_{1:k}) P(T_k = T^l \mid \mathbf{z}_{1:k})}{P(T_{k+1} = T^j \mid \mathbf{z}_{1:k})} \\
&= \frac{\beta(j \mid l) p_k(l)}{\sum_{l'=1}^{N} \beta(j \mid l') p_k(l')}
\end{aligned}
$$

In Eq. (5), the behavioral model term $P(\mathbf{a}_k = \mathbf{a} \mid \mathbf{s}_k = \mathbf{s}^n, \mathbf{z}_{1:k}, T_k = T^l)$ specifies the probability that action $\mathbf{a}$ is taken in state $\mathbf{s}^n$ under intention T^l. The intent-specific state distribution, $P(\mathbf{s}_k = \mathbf{s}^n \mid \mathbf{z}_{1:k}, T_k = T^l)$, represents the state distribution based on the latest data and a specific intention. The next section provides the behavioral model and the recursive computation of this term.

III. Modeling Actions and Intent

This section instantiates the bounded-rational decision-maker formulation in a human-motivated collaborative setting using the structured behavioral models below. We outline the computation of the two key elements in Eq. (5): (1) the behavioral model, which represents the low-level policy under a specific subtask, and (2) the intent-specific state distribution, which represents the posterior distribution of the state given the observation history under a specific subtask.

A. Behavioral Model

1) Optimal Policy under Direct State Observability: Given the existence of N subtasks, the status of each subtask can be represented using a binary vector of size N. A subtask tracker $\eta_k \in \{0,1\}^N$ represents the status of subtasks at time step k, where $\eta_k(j) = 1$ indicates that subtask T^j has already been completed, while $\eta_k(j) = 0$ indicates that subtask T^j has not yet been completed. If subtask T^j is completed at time step k, the jth entry of the subtask tracker switches to 1, i.e., $\eta_k(j) = 1$, and this value remains 1 along the subsequent trajectory.

Under direct state observability, we first consider a rational decision-maker. Behavior under direct state observability can be described using a Markov decision process (MDP) with the augmented state space $\mathcal{S} \times \{0,1\}^N$, action space $\mathcal{A}$, transition probability $p(\mathbf{s}', \eta' \mid \mathbf{s}, \eta, \mathbf{a})$, and reward function $R(\mathbf{s}, \eta, \mathbf{a}, \mathbf{s}', \eta')$. The transition of the subtask tracker is deterministic, given the current subtask tracker and the next state, and is represented by:

$$
\eta'(j \mid \mathbf{s}, \eta, \mathbf{s}') = \begin{cases} \eta(j) & \text{If } \mathbf{s}' \neq \mathcal{G}^j \\ 1 & \text{If } \mathbf{s}' = \mathcal{G}^j \end{cases}, \text{ for } j = 1, ..., N.
$$

It ensures that previously completed subtasks do not contribute to the reward again.

Let $\pi : \mathcal{S} \times \{0,1\}^N \to \mathcal{A}$ represent a policy that maps each augmented state to an action. The optimal policy can be defined as:

$$
\pi^*(\mathbf{s}, \eta) = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}\left[\sum_{t=0}^{h} \gamma^t R(\mathbf{s}_t, \eta_t, \mathbf{a}_t, \mathbf{s}_{t+1}, \eta_{t+1}) \mid \mathbf{s}_0 = \mathbf{s}, \eta_0 = \eta, \mathbf{a}_{0:h} \sim \pi \right],
\tag{6}
$$

for all $(\mathbf{s}, \eta) \in \mathcal{S} \times \{0,1\}^N$; where the expectation is over the stochasticity in the state transition, h denotes the horizon, $\gamma \in [0,1]$ is a discount factor, and the optimization is over all possible policies. For domains with a finite horizon h, the discount factor is typically set to 1, while the discount factor $\gamma \in [0,1)$ is used for infinite-horizon (stationary) cases. The

solution to the optimization in Eq. (6) can be obtained using dynamic programming in finite, small state and action spaces, and reinforcement learning (RL) for large, continuous spaces.

2) Sub-Optimal Reinforcement Learning Behavioral Model: In practice, decision-makers may perceive the environment differently, have partial knowledge of objectives, and exhibit varying levels of proficiency in performing subtasks, making fully optimal decision-making unrealistic. This paper models behavior through a sub-optimal reinforcement learning (RL) agent, where decisions may deviate from the optimal policy defined in Eq. (6). Given the MDP representation of decisions, the behavioral model in Eq. (5) can be expressed as:

$$P(\mathbf{a}_k = \mathbf{a} \mid \mathbf{s}_k = \mathbf{s}, \mathbf{z}_{1:k}, T_k = T^l) = P(\mathbf{a}_k = \mathbf{a} \mid \mathbf{s}_k = \mathbf{s}, \eta_k = \mathbf{e}^l), \tag{7}$$

where e^l denotes the subtask-tracker configuration in which T^l is the only uncompleted subtask: the lth entry is 0 and all remaining entries are 1. The history of observed data is dropped, as the probability is conditioned on the state, and the intention T^l is replaced by the subtask tracker $\eta_k = \mathbf{e}^l$. This subtask tracker $\eta_k = \mathbf{e}^l$ indicates that the policy is focused on pursuing the specific subtask T^l.

We adopt the well-known ϵ-greedy policy [54] for representing the behavioral model in Eq. (7):

$$\begin{aligned}\mu^H(\mathbf{a} \mid \mathbf{s}, \eta = \mathbf{e}^l) &\coloneqq P(\mathbf{a}_k = \mathbf{a} \mid \mathbf{s}_k = \mathbf{s}, \eta = \mathbf{e}^l) \\ &= \begin{cases} \tau + \frac{1-\tau}{|\mathcal{A}|} & \text{if } \mathbf{a} = \pi^*(\mathbf{s}, \eta) \\ \frac{1-\tau}{|\mathcal{A}|} & \text{if } \mathbf{a} \neq \pi^*(\mathbf{s}, \eta) \end{cases},\end{aligned} \tag{8}$$

for $\mathbf{a} \in \mathcal{A}$, $\mathbf{s} \in \mathcal{S}$; where $0 \leq \tau \leq 1$ represents the *low-level rationality parameter*. The superscript in μ^H denotes the instantiated behavioral model used throughout the derivation. The decision-maker is assumed to follow the optimal policy $\pi^*(\mathbf{s}, \eta)$ with probability $\tau + (1-\tau)/|\mathcal{A}|$ and any of the non-optimal actions with probability $(1-\tau)/|\mathcal{A}|$. Thus, the low-level rationality parameter τ near 1 models behavior close to the optimal policy $\pi^*(\mathbf{s}, \eta)$, whereas values close to 0 indicate random decision-making.

The Boltzmann policy [55] is another well-known behavioral model. The behavioral model in Eq. (7) can be expressed according to the Boltzmann model as:

$$\begin{aligned}\mu^H(\mathbf{a} \mid \mathbf{s}, \eta = \mathbf{e}^l) &\coloneqq p(\mathbf{a}_k = \mathbf{a} \mid \mathbf{s}_k = \mathbf{s}, \eta = \mathbf{e}^l) \\ &= \frac{\exp\left(\kappa\, Q^*(\mathbf{s}, \eta, \mathbf{a})\right)}{\sum_{\mathbf{a}' \in \mathcal{A}} \exp\left(\kappa\, Q^*(\mathbf{s}, \eta, \mathbf{a}')\right)},\end{aligned} \tag{9}$$

for $\mathbf{a} \in \mathcal{A}, \mathbf{s} \in \mathcal{S}$; where Q^* represents the state-action value function under the optimal policy π^* in Eq. (6) denoted by

$$Q^*(\mathbf{s}, \eta, \mathbf{a}) = \mathbb{E}\Bigg[\sum_{t=0}^{h} \gamma^t R(\mathbf{s}_t, \eta_t, \mathbf{a}_t, \mathbf{s}_{t+1}, \eta_{t+1}) \mid \mathbf{s}_0 = \mathbf{s}, \eta_0 = \eta, \mathbf{a}_0 = \mathbf{a}, \mathbf{a}_{1:h} \sim \pi^* \Bigg].$$

The parameter $\kappa > 0$ in Eq. (9) represents the low-level rationality parameter, where larger values ($\kappa \to \infty$) model behavior close to the optimal policy, whereas values close to zero represent a random decision-maker.

Both behavioral models in Eq. (8) and Eq. (9) are stochastic and account for possible deviations from the optimal policy. These models are also widely used in inverse learning and imitation learning techniques for modeling decision behavior [45], [48]. The Boltzmann model's softmax action selection is directly grounded in the Luce choice axiom [56] and aligns with resource-rational accounts of decision-making [1], [2]. The ϵ-greedy model provides a simpler one-parameter approximation of systematic deviations from optimal action selection. In Section V, we describe how the rationality parameter can be inferred from the trajectory if a priori knowledge is unknown.

B. Intent-Specific State Distribution

In this section, a recursive computation of the intent-specific state distribution in Eq. (5) is provided. Formally, we define this state distribution for state $\mathbf{s}^i$ and subtask T^j at time step k as:

$$\vartheta_k(i, j) = P(\mathbf{s}_k = \mathbf{s}^i \mid \mathbf{z}_{1:k}, T_k = T^j),$$

which provides a representation of the state conditioned on a specific intention and the observed trajectory. After observing $\mathbf{z}_{k+1}$, the intent-specific state distribution can be updated as:

$$\begin{aligned}\vartheta_{k+1}(i, j) &= P(\mathbf{s}_{k+1} = \mathbf{s}^i \mid \mathbf{z}_{1:k+1}, T_{k+1} = T^j) \\ &= \frac{P(\mathbf{z}_{k+1}, \mathbf{s}_{k+1} = \mathbf{s}^i \mid \mathbf{z}_{1:k}, T_{k+1} = T^j)}{\sum_{\mathbf{s}^r \in \mathcal{S}} P(\mathbf{z}_{k+1}, \mathbf{s}_{k+1} = \mathbf{s}^r \mid \mathbf{z}_{1:k}, T_{k+1} = T^j)} \\ &= \frac{P(\mathbf{z}_{k+1} \mid \mathbf{s}_{k+1} = \mathbf{s}^i) P(\mathbf{s}_{k+1} = \mathbf{s}^i \mid \mathbf{z}_{1:k}, T_{k+1} = T^j)}{\sum_{\mathbf{s}^r \in \mathcal{S}} P(\mathbf{z}_{k+1} \mid \mathbf{s}_{k+1} = \mathbf{s}^r) P(\mathbf{s}_{k+1} = \mathbf{s}^r \mid \mathbf{z}_{1:k}, T_{k+1} = T^j)}\end{aligned}$$

for $\mathbf{s}^i \in \mathcal{S}$ and $j = 1, ..., N$. Similar to Eq. (1), $\vartheta_{k+1}(i, j)$ is in direct proportion to its numerator, which consists of two terms: $P(\mathbf{z}_{k+1} \mid \mathbf{s}_{k+1} = \mathbf{s}^i)$ and $P(\mathbf{s}_{k+1} = \mathbf{s}^i \mid \mathbf{z}_{1:k}, T_{k+1} = T^j)$. For each intention T^j, we shall ensure that $\sum_{\mathbf{s}^i \in \mathcal{S}} \vartheta(i, j) = 1$. The former term is the observation function $\mathcal{O}(\mathbf{z}_{k+1}, \mathbf{s}^i)$, and the latter one can be further unpacked according to Eq. (4) and Eq. (5) as:

$$\begin{aligned}&P(\mathbf{s}_{k+1} = \mathbf{s}^i \mid \mathbf{z}_{1:k}, T_{k+1} = T^j) \\ &= \sum_{l=1}^{N} \sum_{\mathbf{s}^n \in \mathcal{S}} \sum_{\mathbf{a} \in \mathcal{A}} P(\mathbf{s}_k = \mathbf{s}^n, \mathbf{s}_{k+1} = \mathbf{s}^i, \mathbf{a}_k = \mathbf{a}, T_k = T^l \mid T_{k+1} = T^j, \mathbf{z}_{1:k}) \\ &= \sum_{l=1}^{N} \sum_{\mathbf{s}^n \in \mathcal{S}} \sum_{\mathbf{a} \in \mathcal{A}} \mathcal{P}(\mathbf{s}^n, \mathbf{a}, \mathbf{s}^i) \mu^H(\mathbf{a} \mid \mathbf{s}^n, \mathbf{e}^l) \vartheta_k(n, l) \\ &\qquad \frac{\beta(j \mid l) p_k(l)}{\sum_{l'=1}^{N} \beta(j \mid l') p_k(l')}, \text{ for } j = 1, ..., N.\end{aligned}$$

Then we can write ϑ_{k+1} as:

$$\begin{aligned}\vartheta_{k+1}(i, j) \propto \mathcal{O}(\mathbf{z}_{k+1}, \mathbf{s}^i) \sum_{l=1}^{N} \sum_{\mathbf{s}^n \in \mathcal{S}} \sum_{\mathbf{a} \in \mathcal{A}} \mathcal{P}(\mathbf{s}^n, \mathbf{a}, \mathbf{s}^i) \mu^H(\mathbf{a} \mid \mathbf{s}^n, \mathbf{e}^l) \\ \vartheta_k(n, l) \frac{\beta(j \mid l) p_k(l)}{\sum_{l'=1}^{N} \beta(j \mid l') p_k(l')}, \text{ for } \mathbf{s}^i \in \mathcal{S} \text{ and } j = 1, ..., N.\end{aligned} \tag{10}$$

This paper assumes that completion feedback is provided when a subtask terminates. If T^j is completed at state $\mathbf{s}^i = \mathcal{G}^j$, the corresponding intent-specific distribution is set to $\vartheta_{k+1}(i, j) = 1$ and $\vartheta_{k+1}(r, j) = 0$ for all $r \neq i$.

IV. Recursive Intent Update and Subtask-Level Prediction

In addition to estimating the posterior distribution of latent intent, predicting the next subtask-level intent is also important for downstream coordination and decision support under uncertainty. Using the instantiated behavioral model described in Section III as a sub-optimal RL agent, we address both tasks. The following subsections describe updating the intent posterior during subtask execution and predicting the next subtask upon completion. A complexity analysis of the proposed method is also provided.

A. Inferring Latent Intent

When a subtask is being performed, the observation history can be used to update the posterior distribution of latent intent. More specifically, if a new observation $\mathbf{z}_{k+1}$ is obtained and no completion feedback is received, the intent posterior can be updated. This can be achieved by replacing the behavioral model in Eqs. (8)-(9) and the intent-specific posterior in Eq. (10) into Eqs. (1)-(5) for the following recursive posterior update of latent intent:

$$p_{k+1}(j) \propto \sum_{l=1}^{N} \sum_{\mathbf{s}^i \in \mathcal{S}} \sum_{\mathbf{s}^n \in \mathcal{S}} \sum_{\mathbf{a} \in \mathcal{A}} \mathcal{O}(\mathbf{z}_{k+1}, \mathbf{s}^i) \mathcal{P}(\mathbf{s}^n, \mathbf{a}, \mathbf{s}^i) \mu^H(\mathbf{a} \mid \mathbf{s}^n, \mathbf{e}^l) \vartheta_k(n, l) \beta(j \mid l) p_k(l), \text{ for } j = 1, ..., N, \tag{11}$$

where the normalization is over possible latent intents. Despite multiple summations appearing in Eq. (11), we will later show that most elements can be computed offline and in parallel, enabling efficient real-time implementation of this posterior update. The recursion in Eq. (11) is an exact online Bayesian update under the stated model assumptions, namely the fixed candidate subtask set, completion feedback at subtask termination, the behavioral model in Section III, the time- and history-independent switching model, and the known POMDP transition and observation models. The next-subtask predictive prior, introduced in the following subsection, is estimated by Monte Carlo rollouts and is therefore an approximation.

B. Predicting Subtask-Level Intent

Most existing intent recognition, trajectory prediction, and Bayesian inference methods estimate the current intent without explicitly modeling the next subtask-level transition, often assuming a uniform or pre-specified distribution for the remaining subtasks. However, subtask selection may depend on priorities and task geometry; for instance, a rational decision-maker is more likely to pursue a nearby subtask than a distant one with equal priority. Accounting for such high-level rationality can provide informative priors, particularly when a subtask has just been completed.

The behavioral models in Eq. (8) or Eq. (9) represent action-level motion under a candidate subtask, and we leverage them to predict the next subtask-level intent. Assume that subtask T^j has just been terminated at its terminal state $\mathbf{s}_{k+1} = \mathcal{G}^j$. Then, the jth entry of the subtask tracker switches to 1, i.e., $\eta_{k+1}(j) = 1$, and $\eta_{k+1}(i) = \eta_k(i)$ for $i \neq j$. The decision-maker follows the policy $\mu^H(.|\mathbf{s}_{k+1}, \eta_{k+1})$ given the current state and the status of subtasks. More generally, $\mu^H(.|\mathbf{s}, \eta)$ represents the model policy at any state $\mathbf{s}$ and subtask tracker η, indicating rational behavior in achieving the remaining subtasks rather than focusing on specific ones. Since state transitions and the model policy are stochastic, predicting long-term motion toward terminating the next subtask can be challenging; we therefore approximate the next subtask-level intent using Monte Carlo (MC) trajectories.

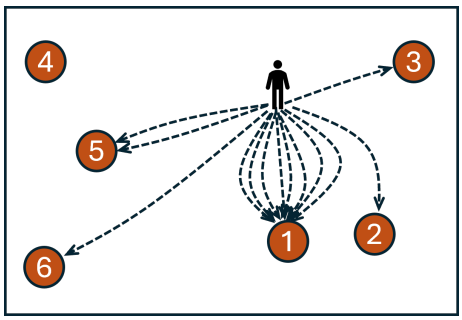


Figure 2: Monte Carlo paths illustrating the probabilistic approximation of the next subtask.

Let $\mathbf{s}_{k+1} = \mathcal{G}^j$ be the current state (i.e., the terminal point of subtask T^j) and η_{k+1} be the subtask tracker (i.e., indicating the remaining subtasks). The next subtask-level intent can be viewed as a prior probability over the next move. The past trajectory is independent of the future subtask-level transition, given the current state and subtask tracker, i.e., $P(T_{k+1} = T^i \mid \mathbf{s}_{k+1} = \mathcal{G}^j, \eta_{k+1}, \mathbf{z}_{1:k}) = P(T_{k+1} = T^i \mid \mathbf{s}_{k+1} = \mathcal{G}^j, \eta_{k+1})$. To calculate this next-move prior, we generate L MC trajectories starting from $\mathbf{s}_{k+1} = \mathcal{G}^j$ and subtask tracker η_{k+1} and following the model policy μ^H as:

$$\left\{\mathbf{s}_t^l, \eta_t^l, \mathbf{a}_t^l \sim \mu^H(.|\mathbf{s}_t^l, \eta_t^l), \mathbf{s}_{t+1}^l, \eta_{t+1}^l, t = k+1, ..., k+1+t^l\right\}_{l=1}^{L}$$

where the superscript l indicates the lth MC trajectory, $\mathbf{s}_{k+1}^l = \mathcal{G}^j, \eta_{k+1}^l = \eta_{k+1}$, and t^l is the first time step that another subtask is achieved in the lth MC trajectory, expressed as:

$$t^l = \{\eta_t^l = \eta_{k+1} \text{ for } k+1 \leq t < k+1+t^l, \text{ and } \eta_{t^l}^l \neq \eta_k\},$$

for $l = 1, .., L$. The trajectories follow the behavioral model corresponding to the current state and the subtask tracker, indicating the propagation of uncertainty in modeled behavior and the state transition probabilities. t^l denotes the time step at which the lth MC trajectory reaches the next subtask. Then, the probability of the next subtask-level intent starting from $\mathbf{s}^j$ under remaining subtask η_{k+1} can be computed using the MC trajectories as:

$$p_{\mathbf{s}^j, \eta_{k+1}}(i) = P(T_{k+1} = T^i \mid \mathbf{s}_{k+1} = \mathcal{G}^j, \eta_{k+1}, \mu^H) \approx \frac{1}{L} \sum_{l=1}^{L} 1_{\eta_{k+1+t^l}(i) \neq \eta_{k+1}(i)}, \text{ for } i = 1, ..., N. \tag{12}$$

Fig. 2 illustrates the Monte Carlo trajectories used to approximate the next-subtask probability distribution formalized in Eq. (12).

Despite considering low-level rationality in creating the MC trajectories, the predictive intent posterior in Eq. (12) represents high-level rationality at the subtask level. In other

words, subtasks that are closer to the current location or have higher priorities (e.g., higher rewards) will have the largest predictive posterior probability. However, decision-makers may not always follow the nearest or most prioritized subtask. This could be due to varied perceptions of priorities, incomplete knowledge of subtasks, the risks associated with different subtasks, and assessments of capabilities. To account for such complexity, this paper models high-level rationality at the subtask level using the following models:

$$T \sim \begin{cases} p_{\mathbf{s}^j,\eta_{k+1}} & \text{w.p. } \alpha \\ Cat[(\mathbf{1}_N - \eta_{k+1})/\|\mathbf{1}_N - \eta_{k+1}\|_1] & \text{w.p. } 1-\alpha \end{cases}, \quad (13)$$

where $0 \le \alpha \le 1$ is the high-level rationality parameter, *Cat* denotes the categorical distribution indicating the probability of the next subtask, $\mathbf{1}_N$ is the unity vector of size N, and $\|.\|_1$ indicates the L_1 norm of a binary vector. With probability α, the decision-maker follows the predictive posterior $p_{\mathbf{s}^j,\eta_{k+1}}$; with probability $1-\alpha$, the next subtask is selected uniformly at random from uncompleted subtasks. Thus $\alpha = 1$ represents fully rational subtask selection, while $\alpha \approx 0$ represents random selection. This hierarchical decomposition is consistent with Bayesian Theory of Mind models that assume approximately rational planning at multiple levels of abstraction [4]–[6]. Such behavioral modeling enables accurate prediction of the next subtask-level intent according to high-level rationality, which can be inferred if initially unknown (details in Section V).

According to Eq. (12) and Eq. (13), the next predictive posterior over intents can be expressed as:

$$p_{k+1} = \alpha\, p_{\mathbf{s}^j,\eta_{k+1}} + (1-\alpha)\,\frac{\mathbf{1}_N - \eta_{k+1}}{\|(\mathbf{1}_N - \eta_{k+1})\|_1}. \quad (14)$$

The predictive posterior efficiently models high-level rationality using a single parameter α. The predictive posterior in Eq. (14) can be seen as a prior probability for the next subtask once a subtask is terminated. This can also be used at the beginning of the process to approximate the initial/prior intent, p_0. This can be achieved by replacing $p_{\hat{\mathbf{s}}_0,\eta_0}$ into Eq. (14), where $\hat{\mathbf{s}}_0 = \text{argmax}_{\mathbf{s}\in\mathcal{S}}\, p(\mathbf{s}_0 = \mathbf{s})$.

Despite the computational cost of generating MC trajectories to approximate the predictive intent posterior probabilities, these computations can be performed offline before execution. For any pair of $\mathbf{s} \in \{\mathcal{G}^1, ..., \mathcal{G}^N\}$ and $\eta \in \{0,1\}^N$, one can use the behavioral model in Eq. (8) or Eq. (9) to generate such MC trajectories and estimate $p_{\mathbf{s},\eta}$. Therefore, the complexity of the predictive high-level posterior does not contribute to real-time computation.

C. Complexity Analysis

The proposed method consists of two main phases: an offline stage that computes the optimal policy under the behavioral model and the corresponding intent priors, and an online stage that performs real-time recursive inference. A full procedural summary is provided in Algorithm 1 (supplementary material). We analyze the computational complexity of each phase below.

Offline Phase. The offline phase consists of two components: (1) computing the optimal policy π^* over the augmented state space, and (2) estimating the prior distribution over latent intent using Monte Carlo rollouts.

For small subtask sets ($N \le 10$), we use tabular Q-learning over the augmented space (s, η), where η tracks completed subtasks. This has worst-case complexity $\mathcal{O}(\text{episodes} \times \text{steps per episode} \times (2^N - 1) \times |\mathcal{S}| \times |\mathcal{A}|)$. To address scalability for large N, one can use a Deep Reinforcement Learning (Deep RL) algorithm that replaces the tabular representation with a function approximator (e.g., neural network). This reduces the complexity to $\mathcal{O}(\text{episodes} \times \text{steps per episode} \times (|\mathcal{A}| + C_{\text{model}}))$, where C_{model} is the per-update cost of the function approximator.

The MC rollout procedure used to estimate the prior intent distribution has complexity $\mathcal{O}(L \times (2^N - 1) \times |\mathcal{S}|)$, where L is the number of simulated trajectories. In practice, this cost is mitigated by selectively generating rollouts from reachable goal states and partially amortizing the computation across online queries.

Online Phase. The online phase performs recursive updates of the intent-specific state distributions (Eq. (10)) and posterior over latent intents (Eq. (11)). Without optimization, the per-step complexity is $\mathcal{O}(N^2 \times |\mathcal{S}|^2 \times |\mathcal{A}|)$, which can become computationally intensive as N and $|\mathcal{S}|$ increase. To reduce runtime, we apply two structural optimizations:

- **Local state restriction:** Updates are limited to a reachable subset $\mathcal{S}_{\text{reach}}(k) \subseteq \mathcal{S}$, determined by short-horizon model dynamics, with $|\mathcal{S}_{\text{reach}}(k)| \ll |\mathcal{S}|$.
- **Active-intent filtering:** Completed subtasks are excluded, maintaining an active intent set of size $N_{\text{act}}(k) \le N$.

With these optimizations, the effective per-step complexity becomes $\mathcal{O}(N_{\text{act}}(k)^2 \times |\mathcal{S}_{\text{reach}}(k)|^2 \times |\mathcal{A}|)$, which is significantly more scalable in practice. These mechanisms preserve inference accuracy while reducing redundant computation and are critical for enabling real-time inference at scale.

V. Discussions and Analysis

The proposed method includes three key parameters that jointly capture a wide spectrum of behavior in the model: τ (or κ) representing low-level rationality, α capturing high-level rationality in subtask selection, and c reflecting persistence in continuing the current subtask once chosen. In this section, we first present the simplified formulation under direct state observability as a special case of the POMDP framework, then examine how the three behavioral parameters model different behavioral modes, and finally discuss how they can be estimated from observed trajectories when unknown a priori. Detailed mathematical derivations and recursive formulations are provided in the supplementary material.

A. Direct State Observability

This paper provides intent inference in POMDP environments, which generalizes inference under direct state observability. Assume the observed data are provided in terms of state, i.e., $\mathbf{z}_{1:k} = \mathbf{s}_{1:k}$. Then, the observation model becomes $\mathcal{O}(\mathbf{z}_k, \mathbf{s}) = 1$, if $\mathbf{z}_k = \mathbf{s}$, and 0 otherwise. In this case, the

recursive computation of the intent posterior in Eq. (11) can be simplified as:

$$p_{k+1}(j) \propto \sum_{l=1}^{N} \sum_{\mathbf{a}\in\mathcal{A}} \mathcal{P}(\mathbf{z}_k, \mathbf{a}, \mathbf{z}_{k+1}) \mu^H(\mathbf{a} \mid \mathbf{z}_k, \mathbf{e}^l) \beta(j \mid l) p_k(l),$$

for $j = 1, ..., N$. Direct state observability removes the state-belief update and the summations over latent states. Computing the intent-conditioned action likelihood terms costs $O(N|\mathcal{A}|)$ per time step. Applying the switching model costs $O(|\mathcal{E}_\beta|)$, where $\mathcal{E}_\beta = \{(l, j) \mid \beta(j \mid l) > 0\}$. The resulting cost is $O(N|\mathcal{A}| + |\mathcal{E}_\beta|)$, which becomes $O(N|\mathcal{A}| + N^2)$ for dense admissible switching and $O(N|\mathcal{A}|)$ for persistent or bounded-degree switching.

B. Impact of Low-Level Rationality

Consider the behavioral model represented in Eq. (8), where τ represents the low-level rationality level. If this parameter τ approaches 1, the modeled decision-maker becomes highly rational in action selection, and the policy μ^H becomes closer to the deterministic policy in Eq. (15) as:

$$\mu^H(\mathbf{a} \mid \mathbf{s}, \eta = \mathbf{e}^l) \to \begin{cases} 1 & \text{If } \mathbf{a} = \pi^*(\mathbf{s}, \eta = \mathbf{e}^l) \\ 0 & \text{If } \mathbf{a} \neq \pi^*(\mathbf{s}, \eta = \mathbf{e}^l) \end{cases}, \tag{15}$$

for $\mathbf{a} \in \mathcal{A}$, $\mathbf{s} \in \mathcal{S}$. In this case, the summation over possible actions is eliminated because highly rational behavior follows a deterministic optimal policy. Intent inference is expected to be more accurate in this case, since the latent intent is more clearly reflected in the observed trajectory.

On the contrary, when τ approaches 0, the modeled decisions become closer to those of a random decision-maker. In this case, the policy assigns equal values to all possible actions, i.e., $\mu^H(\mathbf{a} \mid \mathbf{s}, \eta = \mathbf{e}^l) \to 1/|\mathcal{A}|$ for any $\mathbf{a} \in \mathcal{A}$ and $\mathbf{s} \in \mathcal{S}$. Here, the observed actions and trajectory do not clearly imply an intention toward a specific objective, leading to reduced inference accuracy. Intuitively, latent intent is more recognizable when the trajectory represents a direct path toward a subtask, which becomes less apparent under lower levels of rationality.

C. Impact of High-Level Rationality

This section explores the impact of high-level rationality, represented by the parameter α.

This parameter reflects how the modeled decision-maker chooses the next subtask after completing the current one, with $\alpha = 1$ representing the highest rationality in selecting the order of subtasks. The intent prediction in this scenario can be written, according to Eq. (14), as: $p_{k+1} = p_{\mathbf{s}^j, \eta_{k+1}}$. Conversely, when α approaches 0, the intent prediction can be expressed as: $p_{k+1} = (\mathbf{1}_N - \eta_{k+1})/\|(\mathbf{1}_N - \eta_{k+1})\|_1$, meaning that the next subtask is selected completely at random, leading to additional uncertainty in predicting latent intents.

Note that under our assumption that the feedback is generated immediately when a subtask is completed, the value of α has no impact on the intent-specific state distribution. Since the accurate state is then confirmed, the corresponding distribution can be directly determined.

D. Impact of Persistence

Finally, this section discusses the impact of the persistence parameter c on intent inference. $c = 1$ represents behavior that remains committed to a chosen subtask until its termination, while smaller values allow intermediate switching between subtasks. The intent posterior in this case can be written as:

$$p_{k+1}(j) \propto \sum_{\mathbf{s}^i\in\mathcal{S}} \sum_{\mathbf{s}^n\in\mathcal{S}} \sum_{\mathbf{a}\in\mathcal{A}} \mathcal{O}(\mathbf{z}_{k+1}, \mathbf{s}^i) \mathcal{P}(\mathbf{s}^n, \mathbf{a}, \mathbf{s}^i) \mu^H(\mathbf{a} \mid \mathbf{s}^n, \mathbf{e}^j) \vartheta_k(n, j) p_k(j), \text{ for } j = 1, ..., N,$$

while the intent-specific state distribution can be simplified as:

$$\vartheta_{k+1}(i, j) \propto \mathcal{O}(\mathbf{z}_{k+1}, \mathbf{s}^i) \sum_{\mathbf{s}^n\in\mathcal{S}} \sum_{\mathbf{a}\in\mathcal{A}} \mathcal{P}(\mathbf{s}^n, \mathbf{a}, \mathbf{s}^i) \mu^H(\mathbf{a} \mid \mathbf{s}^n, \mathbf{e}^j) \vartheta_k(n, j), \text{ for } j = 1, ..., N.$$

The summation over the subtasks has been eliminated in this case, indicating that no intermediate switching occurs before the current subtask is completed. Consequently, intent inference in this scenario is expected to be simpler and more accurate. On the other hand, a value of c close to 0 indicates frequent switching between subtasks during execution.

Such intermittent switching makes it difficult to associate observed states with consistent latent intents, thereby increasing the uncertainty of the observer's belief update.

E. Estimating Behavioral Parameters

The proposed intention inference requires knowledge of three behavioral parameters (low-level, high-level, and persistence parameters) to compute the posterior distribution. However, these parameters, functioning as a priori knowledge, may be unknown in practice. In such cases, they need to be estimated alongside the latent intent using the available observed trajectory. Let $\theta = (\tau, \alpha, c)$ or $\theta = (\kappa, \alpha, c)$ represent the parameter vector, depending on which behavioral model in Eq. (8) or Eq. (9) is used. Given that $\mathbf{z}_{1:k}$ is the observed data up to time step k, the maximum likelihood estimate of the behavioral parameters can be expressed as:

$$\hat{\theta}_k = \underset{\theta}{\operatorname{argmax}} \prod_{r=1}^{k} \sum_{j=1}^{N} P(\mathbf{z}_r, T_r = T^j \mid \mathbf{z}_{1:r-1}, \theta). \tag{16}$$

For the Boltzmann behavioral model in Eq. (9), its likelihood in Eq. (16) can be defined as $P(\mathbf{z}_r, T_r = T^j \mid \mathbf{z}_{1:r-1}, \kappa, c, \alpha)$, while for the ϵ-greedy behavioral model in Eq. (8), the likelihood can be written as $P(\mathbf{z}_r, T_r = T^j \mid \mathbf{z}_{1:r-1}, \tau, c, \alpha)$. Both likelihood functions rely on the recursive update of the posterior as:

$$p_r(j) = \begin{cases} \frac{P(\mathbf{z}_r, T_r = T^j \mid \mathbf{z}_{1:r-1}, \theta)}{\sum_{l=1}^{N} P(\mathbf{z}_r, T_r = T^l \mid \mathbf{z}_{1:r-1}, \theta)} & \text{If } \textit{performing a subtask}, \\ \alpha\, p_{\mathbf{s}^j, \eta_r}(j) + \frac{1-\alpha}{\|(\mathbf{1}_N - \eta_r)\|_1} (1 - \eta_r(j)) & \text{Otherwise.} \end{cases}$$

This step includes the high-level rationality parameter α. The solution to the optimization problem in Eq. (16) can be obtained using gradient-free optimization methods, such as kernel-based techniques [57] or Bayesian optimization [58]. The next section provides numerical results regarding the inference of these parameters alongside the corresponding intent posterior.

VI. Numerical Experiments

We analyze the performance of the proposed method in both two-dimensional (2D) and three-dimensional (3D) environments. The bounded-rational decision-maker formulation is instantiated in the human-motivated setting described in the previous sections. All experiments therefore use simulated or LLM-proxy trajectories under this instantiated model rather than real human-subject data; specifically, the LLM-based proxy is prompted to emulate a human decision-maker. These results should be interpreted as methodological validation of the inference framework, where terms such as intent, policy, and ground truth refer to quantities defined under the instantiated bounded-rational model. We consider the following metrics for evaluation:

- Posterior probability assigned to the ground-truth intent under the instantiated model, $p_k(i_k^*)$, where i_k^* denotes the ground-truth intent at time step k;
- Maximum a posteriori (MAP) intent estimate, $i_k^{\text{MAP}} = \text{argmax}_{i=1,\dots,N}\, p_k(i)$, representing the most probable intent under the inferred posterior;
- Decision margin, characterizing how decisively the posterior favors a single intent, computed as the difference between the probabilities of the most probable and the second most probable intents: $C(p_k) = p_k(i_k^{\text{MAP}}) - \max_{i\in\{1,\dots,N\}\setminus i_k^{\text{MAP}}} p_k(i)$, where a larger value indicates a more peaked, less ambiguous inference.

We compare our method against the following representative baselines: *Distance-based Probabilistic method* [29], [34], which approximates the observation likelihood using the Euclidean distance between the observed state and each subtask goal; *Boltzmann-based Probabilistic method*, which models the decision-maker as stochastic yet rational and computes the posterior over intents using the Boltzmann distribution. Since this method requires access to the underlying actions, we provide the true action sequence for benchmarking; *LSTM-based method* [22], a supervised learning approach trained to infer intent from labeled trajectories; and *LLM-based intent inference*, included as an auxiliary reference comparator to test whether general-purpose language reasoning can substitute for structured recursive inference, which queries an LLM at each time step to infer the current subtask intent from partial observation history.

For reproducibility, each LSTM baseline takes the observed state-coordinate window and current subtask tracker as input and predicts the active subtask at the window's final step. It uses a three-layer LSTM with 64 hidden units whose final hidden state is concatenated with the subtask tracker, followed by a 32-unit fully connected layer and a softmax over subtasks. The standard LSTM uses an 8-step window (5-step in 3D), while the dual LSTM switches from a 3-step to an 8-step model at trajectory length 8. Training and evaluation trajectories are generated independently from the simulator, with fresh fixed-length windows drawn each epoch and full trajectories used for evaluation. The models are trained with the Adam optimizer and cross-entropy loss, using learning rates 5×10^{-3} and 3×10^{-3} for the 2D 3- and 8-step models (500 epochs) and 10^{-2} for the 3D model (150 epochs); all hyperparameters were fixed after preliminary tuning.

For evaluation in this experimental setting, we consider two types of trajectories: (1) synthetic trajectories generated from parameterized behavior models with controllable rationality levels, and (2) heterogeneous sub-optimal trajectories generated by an LLM-based proxy agent (OpenAI o4-mini). Here, the LLM is prompted to generate task-selection behavior based on current task progress and remaining subtasks. The LLM modules were executed in zero-shot mode through the official OpenAI API with default settings. Repeated runs under identical conditions yielded comparable trajectory-level patterns, and the reported quantitative results were obtained by averaging across multiple runs to ensure representative, reproducible performance. Further implementation details and full prompt examples are provided in the supplementary material.

A. 2D Grid Environment

Our first experiment is conducted in a 2D grid-world environment, as illustrated in Fig. 3. The environment consists of 10 distinct subtasks, whose terminal states are denoted in orange and numbered from 1 to 10. Yellow cells indicate areas that may pose potential danger or discomfort during task execution. Gray cells represent impassable obstacles. The state space consists of 146 cells, and the available actions are four directional movements, $\mathcal{A} = \{\text{Up, Down, Left, Right}\}$. The motion dynamics are stochastic: the decision-maker moves to the intended neighboring cell with probability ζ, and to one of the perpendicular neighboring cells with probability $(1-\zeta)/2$. Completing a subtask yields a reward of $+100$, entering a yellow cell results in a penalty of -0.9, and each movement prior to subtask completion incurs a penalty of -0.1. This reward structure encourages efficient completion of all subtasks while avoiding yellow cells. Finally, observations are assumed to be noisy. The observed state matches the true state with probability $1-\varsigma$, and otherwise corresponds to one of the four adjacent horizontal or vertical cells with probability $\varsigma/4$, where $\varsigma \in [0,1]$ denotes the noise level.

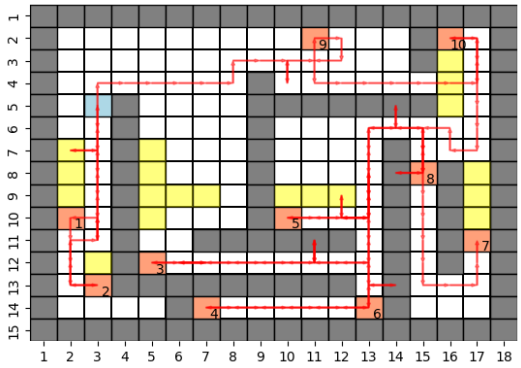


Figure 3: A 2D grid environment with 10 subtasks, where terminal states are marked in orange. Yellow cells represent areas of potential risk, while gray cells denote obstacles that, if crossed, keep the state unchanged.

The following settings are used throughout the experiments unless otherwise specified: low-level rationality $\tau = 0.9$, persistence parameter $c = 0.95$, high-level rationality $\alpha = 0.9$,

discount factor $\gamma = 0.999$, data noise $\varsigma = 0.05$, and stochasticity $\zeta = 0.95$. Q-learning is used for offline learning of the model policy. All average results are obtained from 200 independent runs with the standard error of the mean presented, unless stated otherwise.

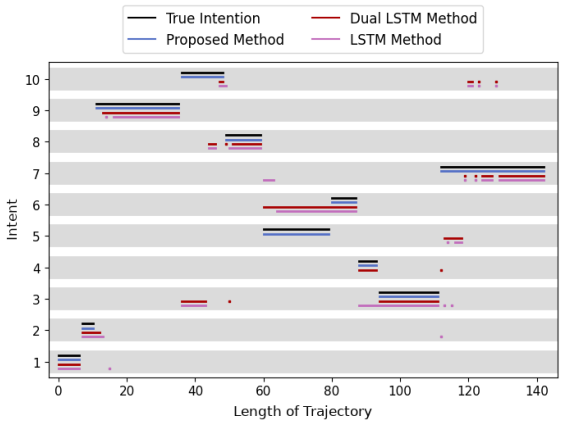


Figure 4: Comparison of the ground-truth intents under the instantiated model with the most probable inferred intents obtained from the proposed method, the standard LSTM-based method, and the dual LSTM variant over time.

Overall Performance Evaluation. In the first experiment, a single trajectory generated under the instantiated bounded-rational model is shown in Fig. 3, with parameters $\tau = 0.9$, $c = 1$, $\alpha = 1$, $\varsigma = 0.05$, and $\zeta = 0.95$. The subtasks are completed in the order: 1, 2, 9, 10, 8, 5, 6, 4, 3, 7. The corresponding ground-truth intents under the instantiated model and the most probable (MAP) inferred intents, produced by our method, a standard LSTM, and a dual LSTM variant, are shown in Fig. 4.

The LSTM and dual-LSTM baselines follow the protocol described at the beginning of Section VI. The proposed method's MAP predictions closely track the ground-truth intent, whereas both LSTM variants show larger discrepancies at the onset of new subtasks. The dual LSTM improves some early transitions (e.g., subtasks 9 and 4) but cannot fully resolve early-stage ambiguity, as LSTMs rely on temporal accumulation that is limited when few observations are available.

The probabilities assigned to the ground-truth intent under the instantiated model, obtained by all methods using 200 trajectories, are presented in Fig. 5. The proposed method achieves higher average performance than the evaluated baselines across the trajectory. In particular, a larger performance margin is observed in the early steps, when fewer trajectory observations are available. This is due to the prediction capabilities of the proposed method, which explicitly incorporates bounded rationality into probabilistic intent inference. Additionally, unlike the other competing methods that do not converge to 1, the proposed method empirically converges to 1. Among the evaluated baselines, the LSTM method achieves higher early-stage performance than the two probabilistic baselines, though it shows slightly lower performance in the final stage. The dual LSTM achieves slightly better overall performance than the standard LSTM, but the difference remains marginal. Notably, all methods show a decline in performance during the initial brief period; this is due to the increased difficulty of predicting the latent intent after completing the first few easily predictable, nearby subtasks.

Fig. 6 demonstrates the average Kullback-Leibler (KL) divergence between the ground-truth intent under the instantiated model and the inferred intent posterior. One can observe that the initial KL divergence for all methods is large due to the availability of larger sets of subtasks and shorter trajectory prefixes. The proposed method achieves lower KL divergence than the evaluated baselines, with values approaching zero. The standard LSTM was used in all subsequent comparisons, as the dual variant showed negligible improvement.

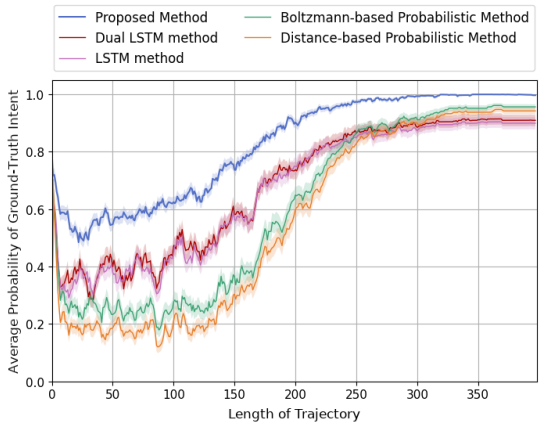


Figure 5: Average probabilities assigned to the ground-truth intents under the instantiated model, obtained by different methods.

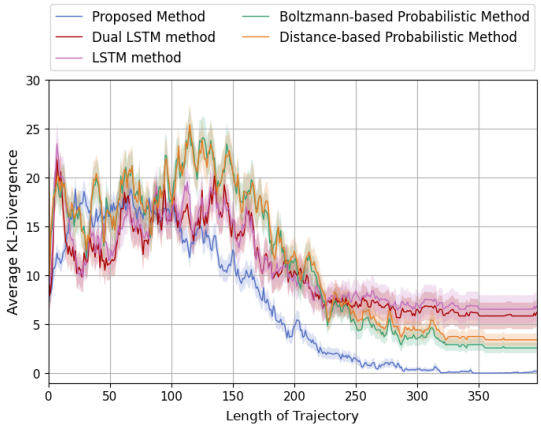


Figure 6: Average KL-divergence between the ground-truth intent under the instantiated model and inferred intent posteriors, obtained by different methods.

Impact of Behavioral Parameters. Subsequently, we investigate the impact of the instantiated behavioral parameters on the proposed method's performance. Fig. 7 demonstrates the average probability assigned to the ground-truth intent under different parameter settings. Each set of bars represents the average performance over multiple simulated trajectories.

In the left bar plot, with $c = 0.95$ and $\alpha = 1$, larger τ values correspond to decision-makers with higher low-level rationality, while smaller values represent more random behavior. The black bars indicate the results of the proposed method. As low-level rationality decreases (i.e., smaller τ), the generated trajectories become less reflective of the ground-truth intent

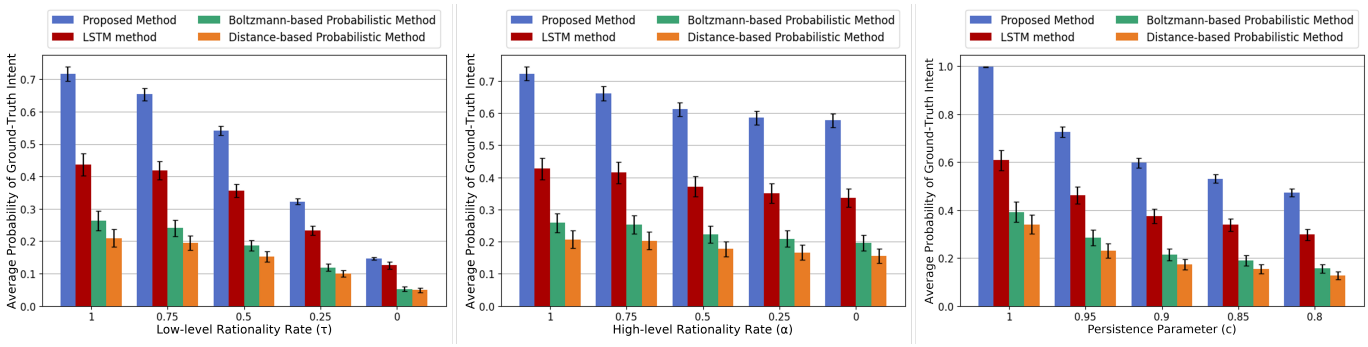


Figure 7: Impact of low-level rationality (left), high-level rationality (middle), and the persistence parameter (right) on the average probability assigned to the ground-truth intent under the instantiated model.

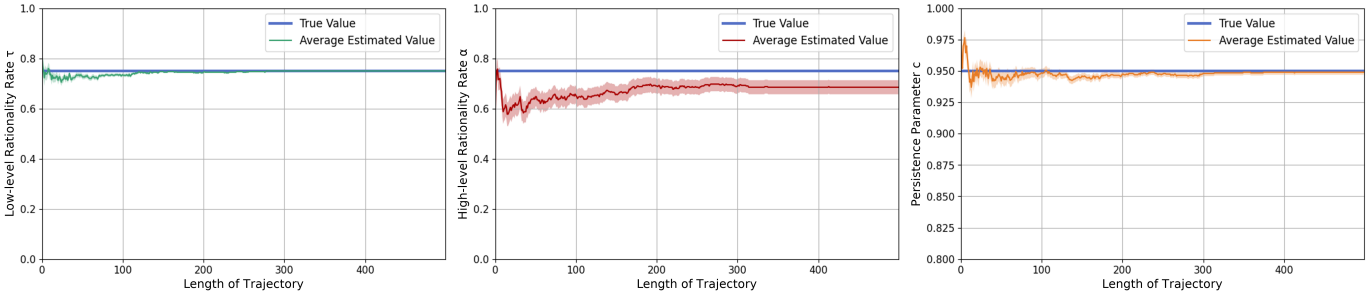


Figure 8: The ground-truth behavioral parameters and the values inferred by the proposed policy: low-level rationality parameter τ (left), high-level rationality parameter α (middle), and persistence parameter c (right).

due to increased randomness in behavior. Consequently, the most accurate intent predictions are achieved for trajectories with the highest low-level rationality. Among all methods, the proposed method consistently performs the best, with the LSTM method showing the second-best results.

The middle plot in Fig. 7, with $c = 0.95$ and $\tau = 1$, represents the impact of the high-level rationality parameter, α. The proposed method achieves the highest average performance among the evaluated methods in every case. In particular, at higher high-level rationality values, the decision-maker selects subtasks more consistently with the model-based prioritization, whereas lower values correspond to increasingly random subtask selection. Similar to previous results, the LSTM method holds the second-best results after the proposed method.

The right plot in Fig. 7 shows the impact of the persistence parameter. We set $\tau = 1$ and $\alpha = 1$, and conducted experiments with c ranging from 0.8 to 1. The highest accuracy is obtained when $c = 1$, representing a decision-maker that does not switch subtasks before completing them. However, as persistence decreases, inference accuracy declines substantially because frequent subtask switching makes the latent intent more difficult to infer. In all cases, the proposed method achieves higher average performance than the evaluated baselines.

These results provide a component-level interpretation of the proposed model. The high-level prior controlled by α contributes most immediately after subtask completion, when within-subtask trajectory evidence is limited and the model-based prior $p_{\mathcal{G}^j,\eta}$ provides an informative distribution over remaining subtasks. The persistence parameter c controls the amount of intermediate subtask switching allowed by $\beta(j \mid l)$, and larger persistence yields a clearer association between observed motion and latent intent. The intent-specific state distribution $\vartheta_k(i, j)$ provides the per-intent belief over the latent state under noisy observations, thereby coupling the POMDP transition and observation models to the intent posterior. Under direct state observability, this per-intent state belief collapses to the observed state, which explains why the state summations disappear in Section V-A. The distance-based and Boltzmann probabilistic baselines do not maintain such a per-intent belief, which helps explain the higher accuracy of the proposed method under noisy partial observations.

We further evaluate the proposed method when the behavioral parameters are not known a priori and must be inferred jointly with the latent intent, using trajectories generated with $\tau = 0.75$, $\alpha = 0.75$, and $c = 0.95$ treated as unknown.

Fig. 8 shows the inferred parameter values obtained by the proposed method. It can be observed that the estimated parameters, especially τ and c, are close to their ground-truth values under the model. The average absolute difference between the ground-truth and estimated intent probabilities, computed using the MAP-inferred parameters, is shown in Fig. 9. The small difference indicates that the proposed method accurately recovers the latent intent defined by the model. In particular, the value converges toward zero for longer trajectories, reflecting improved inference accuracy and fewer remaining unterminated subtasks.

LLM-based Proxy Trajectories and Intent Reasoning. To assess robustness beyond trajectories generated directly by the parameterized simulator, we conduct two additional experiments involving LLM-based proxy trajectories. We use OpenAI o4-mini as a zero-shot proxy generator and observer for heterogeneous sub-optimal decision trajectories.

In the first experiment, the LLM serves as a proxy agent that selects the next subtask after each completion, receiving

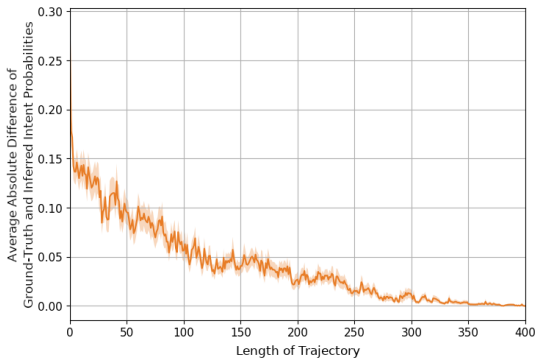


Figure 9: Average absolute difference between ground-truth and inferred intent probabilities, with behavioral parameters simultaneously estimated in Fig. 8, demonstrating the accuracy of the proposed method in intent prediction under unknown behavioral parameters.

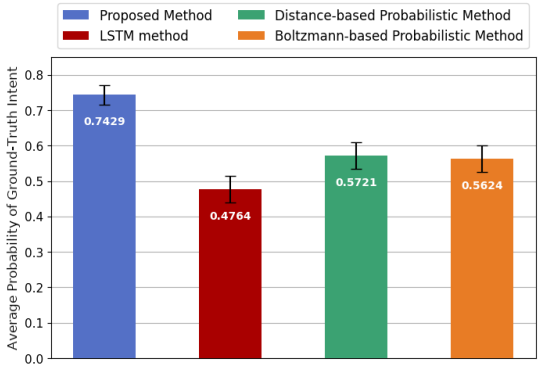


Figure 10: Average probabilities assigned to the ground-truth intent by different methods, evaluated on trajectories generated by the LLM-based proxy agent (OpenAI o4-mini).

the environment layout, subtask status, and behavior pattern as input. We collect 50 LLM-generated trajectories and evaluate all methods. To ensure consistency, the LLM is constrained to commit to each selected subtask.

Fig. 10 shows that our method consistently achieves the highest average probability assigned to the ground-truth intent under the proxy-generated trajectories, demonstrating robust inference on dynamic and sub-optimal behaviors. The LSTM method has lower average performance in this setting, likely due to the complexity and irregularity of LLM-generated patterns. Among the baselines, the distance-based method performs relatively well, as the LLM often selects spatially nearby goals without long-term planning. The proxy trajectories are generated through prompt-conditioned subtask selection, producing a distribution shift relative to the policy-generated synthetic trajectories.

In the second experiment, we evaluate the performance of the LLM-based intent inference baseline. The trajectories used are identical to those in the previous experiments (both synthetic and LLM-generated). For the synthetic trajectories, behavior is generated using the default parameters ($\tau = 0.9$, $\alpha = 0.9$, $c = 0.95$) under the instantiated bounded-rational policy. At each step, the LLM observer receives a partial trajectory and current subtask status, then outputs a probability distribution over the current intent. Environment structure and behavior descriptions are provided at the start of each episode. The prompt was iteratively refined to improve the LLM's reasoning quality. To manage computational cost, we evaluate on representative trajectories that reflect the full distribution.

Table I: Average probability assigned to the ground-truth intent, achieved by different methods, evaluated on two types of trajectories: (1) LLM-generated trajectories from OpenAI o4-mini, and (2) synthetic trajectories generated using the instantiated bounded-rational policy.

Method	LLM-generated data	synthetic data
Proposed	**0.689**	**0.772**
o4-mini	0.451	0.661
LSTM	0.275	0.459
Boltzmann-based	0.431	0.331
Distance-based	0.420	0.267

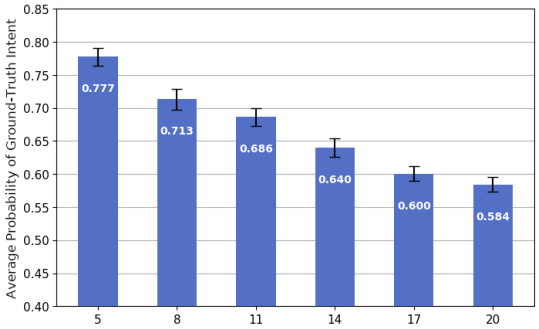


Figure 11: Average probabilities assigned to the ground-truth intent under different numbers of unaccomplished subtasks.

Table I reports the average probability assigned to the ground-truth intent achieved by each method across both data types. Our method outperforms all baselines on both LLM-generated and structured data. On synthetic trajectories, where behavior is more regular, the LLM observer performs reasonably well, as surface-level patterns align with intent. Its performance decreases on LLM-generated data, indicating that prompt-based intent inference alone recovers the proxy agent's latent subtask choices less reliably in this structured environment. The proposed method explicitly maintains the transition model, observation model, and recursive uncertainty update, which supports higher probability assigned to the ground-truth intent across both proxy-generated and synthetic trajectories.

Scalability and Runtime Performance. To evaluate the computational scalability of the proposed framework, we analyze both inference accuracy and runtime as the number of subtasks increases. For large-scale problems where tabular Q-learning is intractable due to the exponential growth of the subtask tracker space, we employ a deep reinforcement learning (Deep RL) formulation to amortize the offline policy computation while preserving the original structure of the learned behavior.

Using a Deep RL model trained on $N = 20$ subtasks in the 2D environment shown in Fig. 3, we vary the number of remaining subtasks from 5 to 20 and measure two key metrics: (1) the average probability assigned to the ground-truth intent, and (2) the average runtime per decision step for updating the intention posterior. Each condition is evaluated across 50 independent trials (maximum 500 steps) using the same trained model, ensuring consistent comparison across N. The results are presented in Fig. 11 and Fig. 12. As shown in

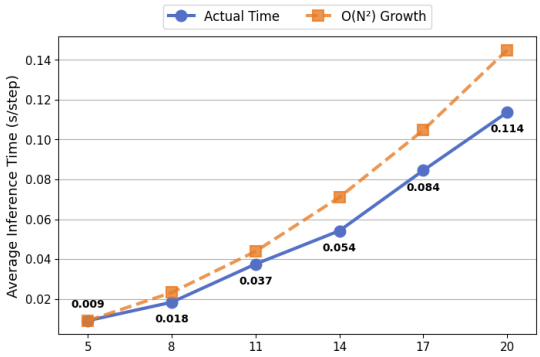


Figure 12: Average inference time per step as a function of the number of unaccomplished subtasks.

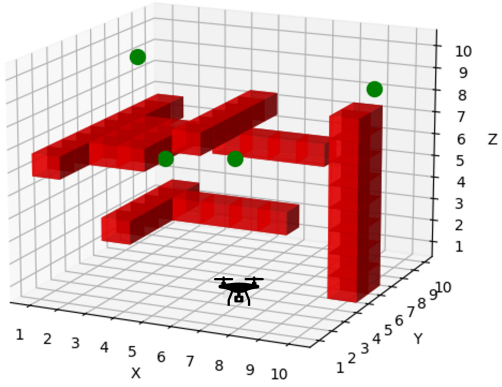


Figure 13: A 3D environment with 4 subtasks, where the initial state is marked with a drone, terminal states are marked with green circles, and obstacles are marked with red blocks.

Fig. 11, the average probability assigned to the ground-truth intent decreases gradually with increasing subtask count. This reflects the expected rise in inference ambiguity as the number of possible goals increases. Despite this, the performance degrades smoothly, and the intent posterior remains stable, indicating robustness of the recursive update against scaling and decision uncertainty.

Since the environment remains fixed, the sizes of the state and action spaces, $|\mathcal{S}|$ and $|\mathcal{A}|$, are constant. The theoretical complexity from Section IV-C thus simplifies from $\mathcal{O}(N^2 \times |\mathcal{S}|^2 \times |\mathcal{A}|)$ to $\mathcal{O}(N^2)$. Fig. 12 confirms this prediction: inference time increases sub-quadratically with N and remains below the theoretical bound. Two practical optimizations, (1) active-intent filtering to reduce the intent set $N_{\text{act}}(k)$, and (2) local state restriction to limit belief updates to $\mathcal{S}_{\text{reach}}(k) \subseteq \mathcal{S}$, substantially reduce unnecessary computation while preserving accuracy. Even in the most demanding case ($N = 20$), the average per-step inference time is approximately 0.11 s on a MacBook Air (Apple M2, 16 GB memory) under single-threaded CPU execution. This runtime is suitable for many task-level collaborative settings; more time-critical applications (e.g., high-speed collision avoidance) would benefit from optimized implementations or deployment on faster hardware. Finally, these results demonstrate the framework's strong computational efficiency and runtime scalability. For comparison, in our implementation the o4-mini-based observer required approximately 30 s per inference step due to prompt length and API latency. These findings emphasize the practical benefits of our symbolic recursive formulation for scalable and interpretable intent inference.

B. 3D Environment

We consider a 3D environment with a drone operating under the bounded-rational decision model as our second setting. As shown in Fig. 13, the environment consists of several obstacles and four subtasks, denoted by green circles. The drone must navigate safely among the obstacles to accomplish all subtasks. Accurate understanding of drone behavior under this model is essential in multi-drone environments to prevent collisions and enable efficient overall coordination.

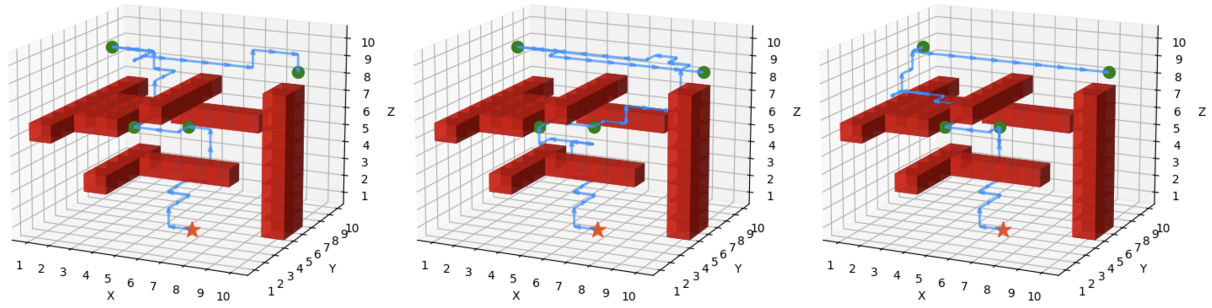
The environment is divided into $10 \times 10 \times 10$ cubes, resulting in 1000 states for the drone. The action set contains six axis-aligned moves in the 3D grid, with state stochasticity represented by $\zeta = 0.95$ as the intended direction and $(1-\zeta)/4$ as one of the other neighboring directions. The data is assumed to be a noisy realization of the drone state, with probability 0.95 as the true state, otherwise one of the 6 possible neighbors. Rewards and Q-learning setup follow the 2D case (+100 per subtask, -0.1 per step). The following parameters are used for our experiments: $\tau = 0.9, \alpha = 0.9, c = 0.95, \gamma = 0.99$.

Fig. 14 shows three trajectories generated under different levels of high-level rationality and persistence. The left plot (a) represents the highest high-level rationality, $\alpha = 1$, reflected in the optimal order of subtask selection under the instantiated model. In the middle plot (b), for a smaller α, the order of subtasks differs from the left plot, as the modeled decision-maker has a higher chance of deviating from the optimal sequence. Comparing the middle and right plots, which reflect the same high-level rationality levels but different persistence parameters, we observe that with smaller persistence (middle plot), the modeled decision-maker is more likely to switch subtasks during operation. This reflects the behavioral variability that bounded-rational decision-makers may exhibit under the instantiated model.

Figure 15 represents the average results obtained by all methods in the 3D environment using 200 generated trajectories. The left plot indicates the average probability assigned to the ground-truth intent. The proposed method achieves substantially higher average performance than the three evaluated baselines, with better convergence. Fig. 15 (b) demonstrates the decision margin for all methods, which indicates the difference between the first and second most probable candidate intents. It can be seen that all methods have similar performance in terms of decision margin, with LSTM having the largest value. However, comparing the results of the left and right plots, we can conclude that the decision margin reflects the confidence of the estimate itself, rather than correctness with respect to the ground-truth intent; a method can be highly confident while still selecting the wrong intent.

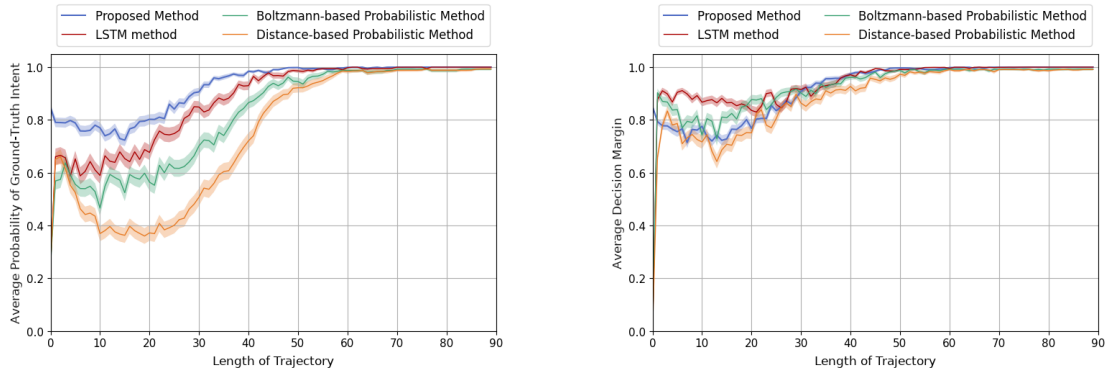
VII. Conclusion and Future Work

This paper presents a probabilistic framework for inferring and predicting latent intentions in partially observable environments, instantiated through a structured behavior model for bounded-rational decision-makers that accounts for suboptimality at both the action and subtask levels. The three



(a) α = 1, c = 0.95 (b) α = 0.9, c = 0.95 (c) α = 0.9, c = 1

Figure 14: Three example trajectories generated in the 3D environment with different levels of high-level rationality and persistence: $\alpha = 1, c = 0.95$ (left), $\alpha = 0.9, c = 0.95$ (middle) and $\alpha = 0.9, c = 1$ (right).



(a) (b)

Figure 15: Comparison of intent inference in a 3D environment, illustrating the performance of the proposed method against competing approaches. (a) Average probabilities assigned to the ground-truth intent, (b) Average decision margin.

parameters of low-level rationality, high-level rationality, and persistence capture key variability in the modeled bounded-rational decision process and can be estimated from observed behavior. Extensive experiments in 2D and 3D environments with multiple subtasks demonstrate that the proposed approach outperforms the evaluated baselines in accuracy, robustness, and adaptability across different rationality levels and observation uncertainties. Moreover, experiments using LLM-generated sub-optimal trajectories further support the framework's effectiveness on diverse proxy-generated behavioral patterns, providing methodological validation of the inference framework.

Future research will aim to extend this framework beyond simulation toward real-world, human–robot collaborative environments. A key direction is conducting IRB-approved human-in-the-loop studies in such physical settings to evaluate the framework with real human trajectory data, which will require systematic participant recruitment and careful experimental design. Additional efforts will investigate settings in which the candidate subtask set may change over time, for example, as subtasks are completed, introduced, or reconfigured. More broadly, future work will study automatic subtask discovery and hierarchical structure learning to relax the assumption of predefined subtasks, as well as methods for intent inference in continuous or high-dimensional state spaces. These directions will broaden the framework's applicability to complex domains such as assistive robotics, autonomous driving, and multi-agent collaboration under uncertainty.

REFERENCES

[1] S. J. Gershman, E. J. Horvitz, and J. B. Tenenbaum, "Computational rationality: A converging paradigm for intelligence in brains, minds, and machines," *Science*, vol. 349, no. 6245, pp. 273–278, 2015.

[2] F. Lieder and T. L. Griffiths, "Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources," *Behavioral and Brain Sciences*, vol. 43, p. e1, 2020.

[3] T. Egner, "Principles of cognitive control over task focus and task switching," *Nature Reviews Psychology*, vol. 2, no. 11, pp. 702–714, 2023.

[4] C. L. Baker, R. Saxe, and J. B. Tenenbaum, "Action understanding as inverse planning," *Cognition*, vol. 113, no. 3, pp. 329–349, 2009.

[5] S. Stacy, S. Gong, A. Parab, M. Zhao, K. Jiang, and T. Gao, "A bayesian theory of mind approach to modeling cooperation and communication," *Wiley Interdisciplinary Reviews: Computational Statistics*, no. 1, p. e1631, 2024.

[6] J. Jara-Ettinger, "Theory of mind as inverse reinforcement learning," *Current Opinion in Behavioral Sciences*, vol. 29, pp. 105–110, 2019.

[7] X. Gao, L. Yan, G. Wang, and C. Gerada, "Hybrid recurrent neural network architecture-based intention recognition for human-robot collaboration," *IEEE Transactions on Cybernetics*, 2021.

[8] A. Ravari, S. F. Ghoreishi, and M. Imani, "Implicit human perception learning in complex and unknown environments," in *2024 American Control Conference (ACC)*, pp. 5453–5459, IEEE, 2024.

[9] S. Ni, L. Zhao, A. Li, D. Wu, and L. Zhou, "Cross-view human intention recognition for human-robot collaboration," *IEEE Wireless Communications*, vol. 30, no. 3, pp. 189–195, 2023.

[10] O. C. Schrempf, D. Albrecht, and U. D. Hanebeck, "Tractable probabilistic models for intention recognition based on expert knowledge," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1429–1434, IEEE, 2007.

[11] Y. Lin, S. F. Ghoreishi, T. Lan, and M. Imani, "Reinforcement learning for Human-AI collaboration via probabilistic intent inference," in *Proceedings of the Reinforcement Learning Conference (RLC)*, September 2025.

[12] Y. Lin, S. F. Ghoreishi, T. Lan, and M. Imani, "High-level human intention learning for cooperative decision-making," in *2024 IEEE*

Conference on Control Technology and Applications (CCTA), pp. 209–216, IEEE, 2024.

[13] L. Zhang, G. Liu, B. Han, Z. Wang, T. Zhang, *et al.*, "semg based human motion intention recognition," *Journal of Robotics*, vol. 2019, 2019.

[14] D. Zhang, L. Yao, K. Chen, S. Wang, X. Chang, and Y. Liu, "Making sense of spatio-temporal preserving representations for eeg-based human intention recognition," *IEEE transactions on cybernetics*, vol. 50, no. 7, pp. 3033–3044, 2019.

[15] Q. Wang, W. Jiao, R. Yu, M. T. Johnson, and Y. Zhang, "Virtual reality robot-assisted welding based on human intention recognition," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 2, pp. 799–808, 2019.

[16] A. Ravari, S. F. Ghoreishi, and M. Imani, "Optimal inference of hidden Markov models through expert-acquired data," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 8, pp. 3985–4000, 2024.

[17] G. Hoffman, T. Bhattacharjee, and S. Nikolaidis, "Inferring human intent and predicting human action in human–robot collaboration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 7, 2024.

[18] W. Wang, R. Li, Y. Chen, Y. Sun, and Y. Jia, "Predicting human intentions in human–robot hand-over tasks through multimodal learning," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 2339–2353, 2021.

[19] B. Liu, E. Adeli, Z. Cao, K.-H. Lee, A. Shenoi, A. Gaidon, and J. C. Niebles, "Spatiotemporal relationship reasoning for pedestrian intent prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3485–3492, 2020.

[20] Y. Cheng, P. Yi, R. Liu, J. Dong, D. Zhou, and Q. Zhang, "Human-robot interaction method combining human pose estimation and motion intention recognition," in *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 958–963, IEEE, 2021.

[21] P. Wu, N. Ahmed, A. Sarma, K. Huang, R. Islam, B. Li, T. Lan, G. Tan, and M. Imani, "Probabilistic verification of cybersickness in virtual reality through bayesian networks," in *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR). Daejeon, South Korea. To appear*, 2025.

[22] M. Mavsar, M. Deniv̈sa, B. Nemec, and A. Ude, "Intention recognition with recurrent neural networks for dynamic human-robot collaboration," in *2021 20th International Conference on Advanced Robotics (ICAR)*, pp. 208–215, IEEE, 2021.

[23] M. Mavsar, B. Ridge, J. Morimoto, A. Ude, *et al.*, "Simulation-aided handover prediction from video using recurrent image-to-motion networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 1, pp. 494–506, 2022.

[24] C. L. McGhan, A. Nasir, and E. M. Atkins, "Human intent prediction using markov decision processes," *Journal of Aerospace Information Systems*, vol. 12, no. 5, pp. 393–397, 2015.

[25] T. Callens, T. Van der Have, S. Van Rossom, J. De Schutter, and E. Aertbeliën, "A framework for recognition and prediction of human motions in human-robot collaboration using probabilistic motion models," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5151–5158, 2020.

[26] O. Dermy, F. Charpillet, and S. Ivaldi, "Multi-modal intention prediction with probabilistic movement primitives," in *Human Friendly Robotics: 10th International Workshop*, pp. 181–196, Springer, 2019.

[27] G. B. Smith, V. Belle, and R. P. Petrick, "Intention recognition with problog," *Frontiers in Artificial Intelligence*, vol. 5, p. 806262, 2022.

[28] S. Jain and B. Argall, "Probabilistic human intent recognition for shared autonomy in assistive robotics," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 9, no. 1, pp. 1–23, 2019.

[29] S. Jain and B. Argall, "Recursive bayesian human intent recognition in shared-control robotics," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3905–3912, IEEE, 2018.

[30] R. Q. Mínguez, I. P. Alonso, D. Fernández-Llorca, and M. A. Sotelo, "Pedestrian path, pose, and intention prediction through gaussian process dynamical models and pedestrian activity recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1803–1814, 2018.

[31] D. Das and C. Lee, "Cross-scene trajectory level intention inference using gaussian process regression and naive registration," tech. rep., Department of Electrical and Computer Engineering, Purdue University, 2018.

[32] R. Singh, T. Miller, J. Newn, E. Velloso, F. Vetere, and L. Sonenberg, "Combining gaze and ai planning for online human intention recognition," *Artificial Intelligence*, vol. 284, p. 103275, 2020.

[33] M. L. Chang, R. A. Gutierrez, P. Khante, E. S. Short, and A. L. Thomaz, "Effects of integrated intent recognition and communication on human-robot collaboration," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3381–3386, IEEE, 2018.

[34] Y. Cheng, L. Sun, C. Liu, and M. Tomizuka, "Towards efficient human-robot collaboration with robust plan recognition and trajectory prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2602–2609, 2020.

[35] H. Berndt, J. Emmert, and K. Dietmayer, "Continuous driver intention recognition with hidden markov models," in *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pp. 1189–1194, IEEE, 2008.

[36] Y. Wang, W. Wan, H. Zhang, C. Chen, and G. Jia, "Pedestrian trajectory intention prediction in autonomous driving scenarios based on spatio-temporal attention mechanism," in *2024 4th International Conference on Electronic Information Engineering and Computer Communication (EIECC)*, pp. 1519–1522, IEEE, 2024.

[37] K. Li, X. Wang, Y. Xu, and J. Wang, "Lane changing intention recognition based on speech recognition models," *Transportation research part C: emerging technologies*, vol. 69, pp. 497–514, 2016.

[38] D. Wang and S. F. Ghoreishi, "Robust reinforcement learning for autonomous driving in uncertain environments," in *2025 IEEE 21st International Conference on Automation Science and Engineering (CASE)*, pp. 1436–1443, IEEE, 2025.

[39] C. Wang, F. Li, Y. Wang, and J. R. Wagner, "Haptic assistive control with learning-based driver intent recognition for semi-autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 425–437, 2021.

[40] K. Vellenga, H. J. Steinhauer, A. Karlsson, G. Falkman, A. Rhodin, and A. C. Koppisetty, "Driver intention recognition: State-of-the-art review," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 602–616, 2022.

[41] D. Wang and S. F. Ghoreishi, "RGDR: Reward-Guided Domain Randomization for Autonomous Driving," in *2025 IEEE 28th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2025.

[42] Z. Huang, H. Liu, J. Wu, and C. Lv, "Differentiable integrated motion prediction and planning with learnable cost function for autonomous driving," *IEEE transactions on neural networks and learning systems*, vol. 35, no. 11, pp. 15222–15236, 2023.

[43] X. Song, K. He, P. Wang, G. Dong, Y. Mou, J. Wang, Y. Xian, X. Cai, and W. Xu, "Large language models meet open-world intent discovery and recognition: An evaluation of ChatGPT," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (H. Bouamor, J. Pino, and K. Bali, eds.), (Singapore), pp. 10291–10304, Association for Computational Linguistics, Dec. 2023.

[44] G. Arora, S. Jain, and S. Merugu, "Intent detection in the age of llms," in *Conference on Empirical Methods in Natural Language Processing*, 2024.

[45] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artificial Intelligence*, vol. 297, p. 103500, 2021.

[46] Z. Zhang, H. Zhou, M. Imani, T. Lee, and T. Lan, "Learning to collaborate with unknown agents in the absence of reward," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, pp. 14502–14511, 2025.

[47] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer, "Hg-dagger: Interactive imitation learning with human experts," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8077–8083, IEEE, 2019.

[48] B. Wang, E. Adeli, H.-k. Chiu, D.-A. Huang, and J. C. Niebles, "Imitation learning for human pose prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7124–7133, 2019.

[49] J. Chen, T. Lan, and V. Aggarwal, "Hierarchical adversarial inverse reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 12, pp. 17549–17558, 2023.

[50] M. Lauri, D. Hsu, and J. Pajarinen, "Partially observable markov decision processes in robotics: A survey," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 21–40, 2022.

[51] H. Kurniawati, "Partially observable markov decision processes and robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 253–277, 2022.

[52] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan, "Cooperative inverse reinforcement learning," *Advances in neural information processing systems*, vol. 29, 2016.

[53] S. Monsell, "Task switching," *Trends in cognitive sciences*, vol. 7, no. 3, pp. 134–140, 2003.

[54] E. Rodrigues Gomes and R. Kowalczyk, "Dynamic analysis of multiagent q-learning with ε-greedy exploration," in *Proceedings of the*

26th annual international conference on machine learning, pp. 369–376, 2009.

[55] S. B. Thrun, *Efficient exploration in reinforcement learning*. Carnegie Mellon University, 1992.

[56] R. D. Luce *et al.*, *Individual choice behavior*, vol. 4. Wiley New York, 1959.

[57] A. Ravari, G. Jiang, Z. Zhang, M. Imani, R. H. Thomson, A. A. Pyke, N. D. Bastian, and T. Lan, "Adversarial inverse learning of defense policies conditioned on human factor models," in *Proceedings of the 57th Asilomar Conference on Signals, Systems, and Computers*, 2024.

[58] M. Pelikan and M. Pelikan, "Bayesian optimization algorithm," *Hierarchical Bayesian optimization algorithm: toward a new generation of evolutionary algorithms*, pp. 31–48, 2005.