2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Risk-Bounded Online Team Interventions via Theory of Mind

Yuening Zhang[1], Paul Robertson[2], Tianmin Shu[3], Sungkweon Hong[1], Brian C. Williams[1]

***Abstract*— Despite advancements in human-robot teamwork, limited progress was made in developing AI assistants capable of advising teams online during task time, due to the challenges of modeling both individual and collective beliefs of the team members. Dynamic epistemic logic has proved to be a viable tool for representing a machine Theory of Mind and for modeling communication in epistemic planning, with applications to human-robot teamwork. However, this approach has yet to be applied in an online teaming assistance context and fails to account for the real-life probabilities of potential team beliefs. We propose a novel blend of epistemic planning and POMDP techniques to create a risk-bounded AI team assistant, that intervenes only when the team's expected likelihood of failure exceeds a predefined risk threshold or in the case of potential execution deadlocks. Our experiments and simulated demonstration on the Virtualhome testbed show that the assistant can effectively improve team performance.**

I. Introduction

When a team works together on a task, its members often coordinate their actions in an impromptu and organic manner. However, real-world teams may not always have a shared understanding of the task. For example, one may be uncertain of others' intent or have false beliefs about what the task requires or how to implement the task. Such misalignment of beliefs can potentially lead to task failure. We envision an AI team assistant that acts as an external observer to the team, intervening when necessary to align team members' beliefs and facilitate the successful execution of the task. Such an assistant informs team members when their beliefs about the task and about each other are false, instructs them on what actions to take, and inquires about what beliefs they hold when the assistant is uncertain itself.

Despite recent advances in human-robot collaboration [1], [2], [3], [4], limited progress has been made in developing AI assistants that oversee teams and offer real-time interventions [5]. This is largely due to the complexity of modeling team members' individual and collective beliefs and the difficulty in defining rich forms of communication that the assistant could employ. Meanwhile, in the field of epistemic planning, dynamic epistemic logic (DEL) emerged as a useful tool to represent an agent's Theory of Mind (ToM) and to model communication between agents, with applications extended to human-robot teamwork [6].

Consider an example task from [6], where a team of two agents, Human and Robot, collaborate to prepare a drink for breakfast. Both agents are equal members who are capable of influencing each other's actions. The task involves the robot selecting a container (either a mug or a glass) and the human choosing a drink (coffee or orange juice), with the success depending on the correct pairing: mug with coffee, and glass with juice.

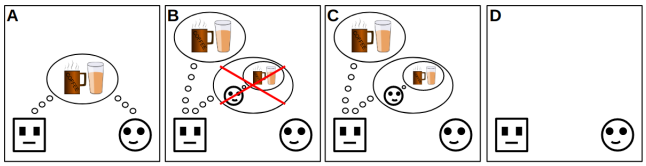

Fig. 1. (A) Robot (square) and Human (circle) share the knowledge of the task constraint. (B) Robot knows the task constraint and correctly knows that Human doesn't know. (C) Robot incorrectly thinks that Human also knows the task constraint. (D) Neither agent knows about the task constraint.

Figure 1 illustrates four possible cases regarding the team's understanding of the pairing constraint. Except for case (A), they illustrate potential misalignment and misconceptions in the team members' beliefs about the task. Among those, an assistant's role becomes vital in the last two scenarios. For example, in case (B), if the robot knows that the human doesn't know about the container-drink pairing, the robot can guarantee success by doing one of the following: wait for the human to grab a drink first and adapt its choice of container, explain to the human about the task constraint, or request the human to take the coffee and take the mug itself [6]. However, in case (C), if the robot is unaware of the human's lack of knowledge, it may grab a container first without considering the possibility of the human choosing an incorrect drink next. It then depends on an external assistant to intervene, either by explaining the task constraint to the human or by informing the robot of the human's lack of knowledge. Moreover, rather than being certain of the team's belief, the assistant may only have a prior belief about the likelihood of the above scenarios.

We introduce *TARS* (**T**eam **A**ssistant with **R**isk-bounded **S**upervision), a computational model for a team assistant. *TARS* infers the team's belief by observing their actions and intervenes when necessary through communication, including asking questions, providing explanations, and announcing intent. To avoid overwhelming the team, the assistant is risk-bounded, intervening only when the team's risk of failure exceeds a predefined risk threshold or to resolve potential execution deadlocks. We show the effectiveness of *TARS* through experiments and demonstration in simulation.

II. Related work

When interaction with other agents is concerned, partially observable Markov decision processes (POMDPs) are a

[1]Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA 01239, USA zhangyn@mit.edu
[2]Dynamic Object Language Labs, Lexington, MA 02421, USA
[3]John Hopkins University, Baltimore, MD 21218, USA

commonly used framework that can model others' hidden internal states [4], [7], [8]. Other agents' behavior is assumed to be conditional on their internal state and stochastic in nature, whose influence on the world is captured by the stochastic transition function. Consequently, these methods typically require pre-specifying or learning an agent behavior model pertaining to the transition function, such as learning an AMM [4]. Building on top of AMM, recent work [5] proposes an AI team assistant that learns a transition model for team members' intended subtasks from past execution data, and intervenes during task time when their inferred subtasks are incompatible. However, it can be challenging to obtain training data when richer mental models are concerned.

Another route without the need for learning is to adopt a nested belief representation – an explicit model of Theory of Mind, such as interactive POMDP (I-POMDP) [9], [10] or social MDP [11], [12]. Assuming that agents are rational and reward-maximizing, each agent can predict others' actions by solving a nested problem according to its belief about their models, before determining its own action. However, these frameworks have limited applicability in practice due to their intractability. It is also impractical to assume accurate prior data on such nested finite probabilistic spaces, and requires a complex model to represent agents' explicit communication with each other that directly alters their beliefs [13].

On the other hand, epistemic planning [14], [15] leveraged epistemic logic to represent a qualitative machine Theory of Mind for planning to achieve epistemic goals. In particular, DEL is used to model how agents' nested beliefs are updated due to both physical and epistemic actions, including communication [14], [16]. Recent work, *EPike* [6], shows the applicability of the approach to human-robot teamwork. Our method is an innovative combination of insights from *EPike* and POMDP techniques to produce a risk-bounded assistant with an explicit Theory of Mind.

III. Problem Formulation

We formulate the assistant's intervention problem based on the CC-POMDP framework [17]. A CC-POMDP builds on POMDP and introduces a *chance constraint* that constrains the system's overall risk of failure to remain below a certain *risk bound*. The team's nested belief is modeled as the hidden state, represented using epistemic logic [18]. We leverage two key ideas from epistemic planning: (1) The state transition model, especially as a result of communication actions, can be defined using DEL. (2) The assistant's observations consist of actions taken by the team, whose probability can be estimated using epistemic planners, assuming that agents are rational. The observations in turn reveal the true belief state of the team, similar to inverse planning [19], [20].

More formally, we define the problem as a DEL-POMDP $\langle \mathcal{S}, b_0, \mathcal{A}, T, \delta, O, C, C_O \rangle$, where $\mathcal{S}$ is a (potentially infinite) set of possible global epistemic states. b_0 is the initial belief state, which is a probabilistic distribution over a finite set of states $S \subset \mathcal{S}$, such as the 4 possible states shown in Figure 1. $\mathcal{A} = \mathcal{A}_I \cup \mathcal{A}_T$ is the set of all epistemic actions partitioned into the assistant's intervention actions $\mathcal{A}_I$ and actions by the agents in the team $\mathcal{A}_T$. $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function. $\delta : \mathcal{S} \times \mathcal{A} \to \{0, 1\}$ is a safety relation that outputs 0 for failure. $O : \mathcal{S} \times \mathcal{A}_T \to [0, 1]$ is the observation probability function that specifies the probability of observing a team action at some epistemic state. $C : \mathcal{S} \times \mathcal{A}_I \to \mathbb{R}$ is the cost function for intervention actions. $C_O : \mathcal{S} \times \mathcal{A}_T \to \mathbb{R}$ is the cost function for observations.

Note that our definition differs from the standard POMDP formulation in two ways: (1) As in CC-POMDP, the formulation introduces a safety relation, permitting certain state-action transitions to terminate in a failure state. For example, we consider it a failure if the assistant explains to an agent something that is not actually true, or if a team member takes an action that leads to the failure of the task. As a result, the formulation requires an additional specification of a risk bound Δ, the maximum allowed risk of failure. (2) Since an observation is also an action taken by the team, it provides an additional state transition, incurring further cost and potentially resulting in failure as well. Lastly, we examine the problem within a finite-horizon context, defining h as a finite horizon.

We denote H_t as the action-observation history up to time t, where $H_t = [a_0, o_1, a_1, o_2, ..., a_{t-1}, o_t]$. Since both intervention and observations are actions in $\mathcal{A}$ that trigger state transitions, the state trajectory is an unobserved sequence $[s_0, s'_1, s_1, s'_2, s_2, ..., s'_t, s_t]$, such that $b_0(s_0) > 0$, $T(s_{k-1}, a_{k-1}, s'_k) > 0$, $O(s'_k, o_k) > 0$, and $T(s'_k, o_k, s_k) > 0$ for $k = 1, ..., t$. A state trajectory fails at time t if $\delta(s_{t-1}, a_{t-1}) = 0$ or $\delta(s'_t, o_t) = 0$. Note that once we reach a failure state, it does not matter what happens next. In other words, a failure state can be considered a special absorbing state that incurs no further cost.

The solution to the finite-horizon DEL-POMDP is a policy π that maps an action-observation history H_t to intervention actions to take. When deterministic policies are concerned, the output is simply an action $a_t = \pi(H_t)$. When stochastic policies are allowed, it specifies a probability distribution of actions to take. For this work, we experimented with solvers that allow both types of policies, as described in Section IV.

An optimal solution π^* is one that minimizes the expected cumulative cost for the finite horizon h:

$$\pi^* = \arg\min_{\pi} \mathbb{E}\big[\sum_{t=0}^{h-1} C(s_t, a_t) + C_O(s'_{t+1}, o_{t+1}) \big| b_0, \pi\big] \quad (1)$$

subject to the chance constraint:

$$1 - Pr\big(\prod_{t=0}^{h-1} \delta(s_t, a_t)\delta(s'_{t+1}, o_{t+1}) = 1 \big| b_0, \pi\big) \leq \Delta \quad (2)$$

where $Pr(\prod_{t=0}^{h-1} \delta(s_t, a_t)\delta(s'_{t+1}, o_{t+1}) = 1)$ is the probability of no failure occurring in the state trajectory.

A. Interleaving Interventions & Observations

The action-observation history H_t naturally corresponds to the assistant and the team taking turns to act. However, it is reasonable to expect that the assistant can choose not to intervene or to intervene consecutively if needed, such as

asking a question to one agent before explaining to another, or providing multiple explanations consecutively. Therefore, in our modeling of DEL-POMDP: (1) we introduce a special $noop \in \mathcal{A}$ as an option for the assistant not to intervene, and (2) we introduce a special $skip \in \mathcal{A}_T$ that represents the skipping of the team action immediately following a non-noop intervention. Formally, if $a_t \neq noop$, then $o_{t+1} = skip$. This way, agents only get a chance to act if the assistant has completed all intended interventions. Note that this assumes that the assistant can always intervene in time if it needs to before agents' actions. For any state s, $T(s, skip, s) = 1$, $\delta(s, skip) = 1$, and $C_O(s, skip) = 0$.

B. State Transitions via DEL

The primary advantage of DEL-POMDP is its utilization of DEL for defining the transitions of the team's belief state in response to both physical and communication actions. In this section, we describe the general framework pertinent to any DEL formulation. The next section discusses our specific choice of DEL concerning multi-agent task execution [6].

We begin by defining the state space $\mathcal{S}$ for DEL-POMDP. Each state $s \in \mathcal{S}$ is a global epistemic state represented by a pointed Kripke model [18], which encompasses not only the actual state but also agents' nested beliefs about the state. We require each epistemic state to be *global*, that is, it fully determines what beliefs each agent holds about the state and about each other. The assistant may be uncertain about the true epistemic state, and its uncertainties are captured by the initial belief state b_0 – a probability distribution over global epistemic states.

For the action space $\mathcal{A}$, including both intervention actions and observations of team actions, each action is an epistemic action represented by a pointed action model with a similar Kripke structure [21]. In this work, we focus on deterministic actions. An action a updates a state s via the *product update* $s' = s \otimes a$. Actions also have *preconditions* that determine their *applicability* in a specific state s. Denoting $\mathcal{A}_I(s)$ as the set of all applicable interventions in $s \in \mathcal{S}$, the set of possible interventions in belief state b with domain S includes $\mathcal{A}_I(s)$ for all $s \in S$, in addition to some question-asking actions derived from the set S directly, described in Section III-C.2.

Following *EPike* [6], we assume that agents act asynchronously instead of taking joint actions simultaneously as in multi-agent MDPs [22]. Thus, a team action refers to any individual agent's action within the team. The presented formulation assumes the availability of an observation function O, which outputs the list of possible team actions along with their probabilities $O(s, o)$, for any given epistemic state s. However, such O may not be readily accessible. In our work, we first obtain function O_i for each agent i that predicts the probability of their actions. This function O_i can be estimated using existing techniques in epistemic planning [6], [16]. Subsequently, we calculate the overall observation function O from the individual functions O_i.

More specifically, when considering the team action as a whole, two things may occur: either one of the agents takes an action first or none of the agents act, resulting in an execution deadlock where no progress is being made. Denoting inaction as $noop$, the team only takes a $noop$ if none of the agents act. Thus, the probability of a team-wise $noop$ in state s is computed as $O(s, noop) = \prod_{i \in Ag} O_i(s, noop)$. For a non-$noop$ action, its probability can be computed with the assumption that there is uniform probability for each agent's non-$noop$ action to occur first in an asynchronous setting, given a specific combination of actions chosen by the agents.

Finally, we define two task-specific concepts. First, we define a set of *successful* epistemic states G, where $s \in G$ denotes a state that reaches the team's end goal. Such a goal may be the team having successfully completed the task, or it can also include epistemic goals, such as when all the team members know that they have completed the task. Second, we define a set of *failure* epistemic states F, where $s \in F$ is a state that reaches a failure state of the task, such as if the combination of mug and juice is selected.

Given the above, we provide the following specifications for the DEL-POMDP functions:

- $\delta(s, a) = 1$ if and only if action a is applicable in s and $s' = s \otimes a \notin F$.
- $T(s, a, s') = 1$ if and only if a is applicable in s and $s' = s \otimes a$ and 0 otherwise.
- $O_i(s, o_i) > 0$ only if agent i considers action o_i to be applicable in s from its local perspective.
- $C(s, a) > 0$ for $a \neq noop$ and $C(s, noop) = 0$. That is, any non-$noop$ intervention action should have a cost.
- $C_O(s, noop) > 0$ if $s \notin G$, and 0 otherwise. That is, there is a cost to the team taking no action when the task has not succeeded. For the rest of the team actions, for this paper, we assume $C_O(s, o) = 0$.

The inclusion of a chance constraint concerning the safety relation allows the assistant to take some risks. For example, the assistant may not need to intervene if the likelihood of the team taking incorrect actions is small. When the assistant is almost certain about an agent's belief, it may also intervene directly based on the information, such as announcing the agent's belief to another agent, without asking the agent first to confirm.

C. DEL for Multi-Agent Task Execution

To model interventions for multi-agent task execution, our DEL formulation follows from the work of *EPike* [6]. More specifically, we leverage a variant of epistemic logic, called the conditional doxastic logic for knowledge bases (CDL-KB), whose semantics are defined by plausibility models [23] instead of Kripke models. At a high level, such a logic differs from the most commonly seen epistemic logic in two ways: (1) Instead of describing agents' beliefs regarding the state of the world, we describe their beliefs regarding a knowledge base that encodes their knowledge of the task, that is, what plans are feasible. (2) Conditional doxastic logic (CDL) pre-encodes how agents' beliefs get revised upon new, potentially contradicting, evidence. This allows us to model agents with false beliefs and allow the assistant to intervene and correct their false beliefs. We focus next on how to model the role

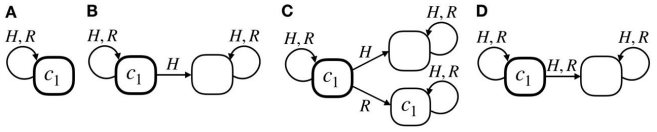

Fig. 2. Epistemic states represented by pointed plausibility models [6]. H denotes the human. R denotes the robot. Each node is a possible world, representing a knowledge base that captures the feasible plan space of the task. c_1 is the constraint associated with the container-drink pairing. The highlighted world represents the true world, also known as the *pointed* world. The arrows represent the order of plausibility regarding the worlds from the agents' perspectives. In (B), the single arrow pointing to the right (with no reverse arrow) labeled with H indicates that the human considers the right world to be more plausible than the left; hence in the human's belief, c_1 is not an explicit constraint for the task.

of an assistant in the framework, while referring the readers to [6] for the details of CDL-KB and the task representation.

1) Belief State: The assistant's belief state is a probability distribution over the set of epistemic states, each specifying a potential nested belief scenario of the team. These epistemic states are represented by pointed plausibility models as referenced in *EPike* [6]. For example, Figure 2 shows the models that correspond to the different cases in Figure 1. While the assistant's belief is probabilistic, each epistemic state is qualitative and does not contain probabilities. Additionally, agents in the team do not actively consider the assistant's participation in the task.

2) Intervention Actions: In *EPike*, an agent can perform four types of actions, namely, execution actions, explanation actions, intent announcements, and question-asking actions [6]. For our assistant, it intervenes via communication only and hence can take all but execution actions.

Explanation & Intent Announcement: An explanation action allows one to communicate their beliefs about the task or their understanding of others' beliefs to those who may have uncertain or false beliefs. For example, the assistant may privately explain to the robot that the human does not know the container-drink pairing. An intent announcement allows one to commit the team to a specific choice of actions. For example, the assistant may request the human to take juice. In *EPike*, the precondition for agent i to explain φ is that agent i must believe what it explains is true, i.e. $B_i\varphi$. Similarly, for agent i to announce the intent c, the precondition is that agent i believes the intent is satisfiable, i.e. $B_i sat(c)$. These preconditions also dictate what the rest of the team observes as the announced truth – not that φ or $sat(c)$ is actually true, but that the actor, agent i, believes them to be true. When an assistant takes the above two types of actions, we assume the assistant's words are always taken as the truth by the agents. Therefore, in the assistant's action model, we simply remove any prefix of B_i from the precondition.

Asking Questions: Agents can ask questions about others' beliefs concerning φ. For example, one might ask another agent whether their intent is coffee or not, or whether they know about a certain task constraint. In DEL, the possible answers of *yes*, *no*, and *I don't know* are modeled by the possible events of an action model [21]. Within the POMDP model, we can model the effect of an assistant's question by modeling the agent's answer as the observation o_{t+1} received immediately after the question is asked a_t. Each possible observation o corresponds to an answer to the question being announced privately, modeled as a private announcement action [21]. $O(s, o) = 1$ iff $s \vDash pre(o)$, where pre denotes precondition.

3) Defining Success & Failure: For this paper, concerning the problem of multi-agent task execution, we define success as the team completing the execution of a feasible plan for the task. We define failure as when there are no feasible plans that remain for the task.

With the above formulation, an intervention will not result in failure as long as it is applicable in state s. However, since the assistant has uncertainty regarding the true s, it may still take an inapplicable intervention. Additionally, because the precondition for an agent's action depends only on the agent's subjective belief, an action is applicable in s from the agent's local perspective if and only if it is actually applicable in s. This means that an observed action inapplicable in s provides evidence to the assistant that s is not the true state.

D. Estimating Probability of Observations

To provide an estimation of O_i for each agent i, we assume that agents are rational and will not take an action that they consider to be inapplicable. We take advantage of the online planning algorithm of *EPike* that outputs a list of utility-maximizing actions for an agent [6] in a state s, and assumes that an agent has a uniform probability of taking any action from the set. Since the algorithm is based on Monte Carlo Tree Search (MCTS), a timeout is set for how long the assistant runs the algorithm to estimate each agent's actions.

IV. Risk-Bounded Algorithms

To find a policy π that meets the risk bound Δ, we first describe how to compute the overall risk of failure and the expected total cost for a policy π. We then describe how we adapt three state-of-the-art solvers for variants of CC-POMDP to solve DEL-POMDP.

A. Calculating Risk of Failure

To calculate the overall risk of failure for a policy, we follow the CC-POMDP formulation [17]. The assistant counts the probability of a transition that violates the safety relation towards the risk spent, and continues acting conditional on no failure having occurred. As outlined in [17], we use $er(b_t|\pi)$ to represent the execution risk of policy π from belief state b_t for the remainder of the planning horizon. We further define safe belief $\tilde{b}_t$ as the belief at time t conditional on all states being safe, that is, the leading transitions to the state all satisfy the safety relation δ. We assume the initial belief to be safe: $\tilde{b}_0 = b_0$. Eq (2) then amounts to satisfying the condition $er(\tilde{b}_0|\pi) \leq \Delta$. The calculation of $er(\tilde{b}_t|\pi)$ and the expected total cost for a DEL-POMDP policy must take into account an additional transition provided by the observed team action on top of the calculation in [17]. A single-step expansion of the DEL-POMDP explicit decision tree is shown in Figure 3, where each node keeps track of the safe belief given the corresponding action-observation history.

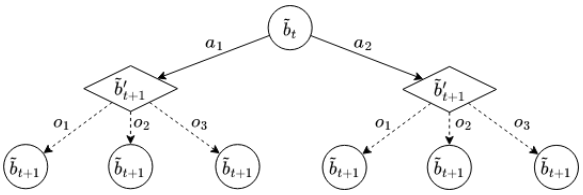

Fig. 3. Single-step expansion for the DEL-POMDP explicit decision tree

B. CC-POMDP Solvers

RAO* [17] is a risk-bounded version of AO* designed for CC-POMDPs. To efficiently prune the policy space that violates the chance constraint, it maintains an upper risk bound for each node – the maximum amount of risk that can be spent by the node going forward, and prunes any branches whose estimated lower bound of execution risk exceeds the upper bound. Such pruning allows RAO* to be very efficient at a cost of suboptimality, because it allocates risk greedily among the different possible observations. We also consider a variant, *safe RAO**, that tries to ensure that the execution risk at all time steps remains below the overall risk bound. To modify RAO* for our purpose, we make sure that it computes the safe belief, execution risk, and the expected total cost at each node in Figure 3 as defined in Section IV-A.

ACDC [24] is an anytime optimal solver for constrained POMDPs (CPOMDPs), producing deterministic policies. A CC-POMDP can be considered as a CPOMDP with only one constraint, namely the chance constraint. ACDC solves the problem in two phases. In the first phase, it solves a dual problem to quickly find a satisficing solution, and in the second phase, it systematically explores deviating policies from the current incumbent to improve the quality of the solution until the optimal one is found. To modify ACDC for our purpose, we can emulate the expansion in Figure 3 by introducing dummy branches of observations that lead to a terminal failure state, used to encode the probability of failure resulting from an inapplicable intervention action or observations of team actions that lead to failure.

CC-POMCP [25] is an MCTS algorithm for CPOMDPs that extends the idea of POMCP [26] to constrained cases. It produces stochastic policies and converges to the optimal policy over time. To handle the constraints, it uses the dual formulation to convert the problem into an unconstrained one, and optimizes the value of the dual parameter at the same time as the policy. Since MCTS only requires a black box simulator that generates a sample of $(s_{t+1}, o_{t+1}, c_t, er_t)$ given (s_t, a_t), where c_t and er_t are the cost and risk spent at a single step, we can simply implement such a simulator function that emits samples according to Section IV-A.

To run the assistant online, there are two options. In the case that the system is guaranteed to terminate within h steps, where h is the horizon, the assistant can simply follow the policy found by our risk-bounded algorithm, such as RAO*. If not, we can consider a receding-horizon control strategy, where the assistant only takes the first action of the policy at each step, and replans upon each observation. This may be a reasonable option as we often want to limit the horizon h for efficiency, or if we cannot predict when the system will terminate. In the case of receding-horizon control, the assistant keeps track of the probability that execution is still safe P_{sa}. Every time an intervention action a_t is taken and an observation o_{t+1} is received, the current belief is updated, with P_{sa} updated accordingly. The assistant replans every time using the updated risk bound $(\Delta - (1 - P_{sa}))/P_{sa}$ that discounts the risk already spent.

V. Experiment Results

To show the effectiveness of our assistant, we evaluate the success rate of the team in executing a task with and without the assistant and demonstrate the assistant in simulation. For the experiments, we set horizon $h = 3$ and use a receding-horizon strategy for execution. We set the intervention cost for explanations to 1, question-asking and intent announcement to 2, and the cost for team inaction to 1. To estimate $O_i(s, o_i)$ for each agent i, *EPike*'s MCTS algorithm is run for a maximum of 100 iterations and with a timeout of 0.2 seconds, whichever comes first. To sample an observation of team action at each time step, we first determine each agent's action by running *EPike*'s MCTS algorithm for 500 iterations without timeout, then randomly select a non-$noop$ action from the team with uniform probability. The experiments were run on an Intel Core i7-6700 running at 3.40GHz.

Success Rate: We first evaluate the success rate of multi-agent task execution on a set of randomly generated sequential tasks. We consider a 2-agent team taking turns to act, with 2 choices per action, similar to picking a mug or a glass. Task constraints of the form "if this choice of action is taken, then that other choice of action must be taken" are randomly sampled and added to the task, while making sure that the task has at least a feasible plan. We sample from 1 to 5 actions per agent, and from 1 to 3 task constraints that the agents are missing collectively, that is, some agents may not know the constraint. For each missing constraint, a probabilistic distribution of two out of four cases in Figure 2 is assigned. That is, for each constraint, the assistant considers it possible that (1) both agents know the constraint, (2) one doesn't know the constraint but the other agent knows that it doesn't know, (3) one doesn't know the constraint and the other mistakenly thinks that it does, and (4) neither agent knows. More specifically, we include test cases with pre-assigned distribution of $\{0.5, 0.5, 0, 0\}$, $\{0.5, 0, 0.5, 0\}$ and $\{0.5, 0, 0, 0.5\}$ for the above four cases, and also with randomly generated distribution such as $\{0, 0.24, 0, 0.76\}$. The above results in 60 randomly generated test cases.

We evaluate the success rate and the total cost under a timeout of $[0.5, 2, 5, 10]$ seconds, using a risk bound from $[0.1, 0.2, 0.3, 0.5]$. If no solution is found within the timeout, no intervention is taken. With a total of 60000 trials, Figure 4 top row shows the averaged result for the generated test cases over all the risk bounds for the different risk-bounded algorithms. The results show that the assistant effectively improves the team's success rate when given enough time,

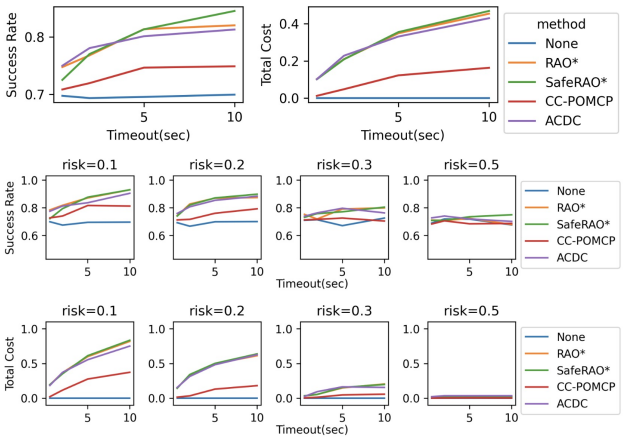

Fig. 4. Average success rate and total cost with and without assistant

with safe RAO* resulting in the highest success rate and CC-POMCP resulting in the lowest intervention cost, yet also the lowest success rate. The bottom two rows show the average success rate and total cost under different risk bounds. We see that both the success rate and the total cost decrease as we increase the risk bound. In particular, when the risk bound is 0.5, the total cost of intervention is close to 0, that is, the assistant barely intervenes. This demonstrates that the assistant is risk-bounded, whose risk tolerance can be adjusted to be less invasive.

Demonstration: To illustrate the types of Theory of Mind scenarios that can be modeled in this framework and how the assistant can help the team, we provide demonstrations in 2 hand-crafted domains. For the demonstration, we use the RAO* algorithm. First, we consider the domain described in the Introduction shown in Figure 1. We integrate the *EPike* algorithm [6] and our assistant, *TARS*, with the Virtualhome simulator [27], and demonstrate 2 scenarios detailed in the attached video. Second, we consider a simple Search-and-Rescue domain, where two agents must put out fire, if there is one, before rescuing a victim in the room. They need the medkit to rescue the victim and the extinguisher to put out the fire. Both agents are capable of performing all the actions. We assume the assistant can only communicate privately with each agent. We run the algorithm on 3 scenarios and illustrate the results in Figure 5. Note that in scenario (B), we describe a situation where actions are partially observed by agents in the team and the assistant can explain the occurrence of an action to the agents. This is a novel feature compared to *EPike* that assumes agents have full observability of actions, since as an external observer, the assistant can be reasonably assumed to have full observability of action execution.

Compared to an assistant formulated as a third agent in the team using the *EPike* formulation, our assistant has the benefit that: (1) It assigns probabilities to the different possible epistemic states, and takes a risk-bounded approach to be less invasive. (2) It assumes a higher priority than regular agents in the team, that is, its interventions are assumed to precede any team actions, and its words are trusted by the team. (3) Its belief state is updated through inverse planning using

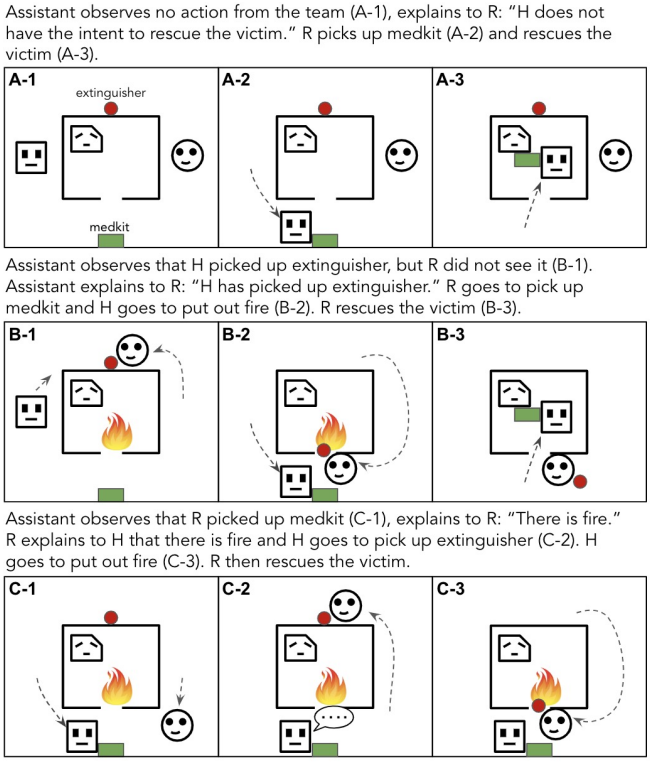

Fig. 5. A robot (square) and a human (circle) need to put out the fire and rescue the victim: (A) Both agents expect that the other agent will rescue the victim. (B) The robot did not see that the human had picked up the extinguisher. (C) Neither agent knows that there is a fire.

Bayesian inference upon observation of team actions, which is stronger than *EPike*'s belief update that relies on DEL action update only. Our work also has limitations, including: (1) Similar to [6], the knowledge base for the task considers only unconditional feasible plans for the task and does not handle contingent world state observations. Future work can incorporate agents' beliefs about the current world state [21] and consider richer plan representations, such as conditional plans, in agents' beliefs about the task. Second, it requires the initial pool of possible epistemic states to be specified. Future work can investigate the generation of epistemic state hypotheses on the fly [3].

VI. Conclusion

In this work, we combine insights from epistemic planning and POMDP techniques to develop a risk-bounded AI team assistant that improves teamwork through interventions. We validated through experiments and simulation that the assistant is effective in improving team performance.

VII. Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001120C0035. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA).

REFERENCES

[1] S. J. Levine and B. C. Williams, "Watching and acting together: Concurrent plan recognition and adaptation for human-robot teams," *Journal of Artificial Intelligence Research*, vol. 63, pp. 281–359, 2018.

[2] F. Semeraro, A. Griffiths, and A. Cangelosi, "Human–robot collaboration and machine learning: A systematic review of recent research," *Robotics and Computer-Integrated Manufacturing*, vol. 79, p. 102432, 2023.

[3] X. Puig, T. Shu, J. B. Tenenbaum, and A. Torralba, "Nopa: Neurally-guided online probabilistic assistance for building socially intelligent home assistants," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 7628–7634.

[4] V. V. Unhelkar, S. Li, and J. A. Shah, "Decision-making for bidirectional communication in sequential human-robot collaborative tasks," in *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 2020, pp. 329–341.

[5] S. Seo, B. Han, and V. V. Unhelkar, "Automated task-time interventions to improve teamwork using imitation learning," in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, N. Agmon, B. An, A. Ricci, and W. Yeoh, Eds. ACM, 2023, pp. 335–344.

[6] Y. Zhang and B. Williams, "Adaptation and communication in human-robot teaming to handle discrepancies in agents' beliefs about plans," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 33, no. 1, 2023, pp. 462–471.

[7] X. Huang, S. Hong, A. Hofmann, and B. C. Williams, "Online risk-bounded motion planning for autonomous vehicles in dynamic environments," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 29, 2019, pp. 214–222.

[8] 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.

[9] "A framework for sequential planning in multi-agent settings," *Journal of Artificial Intelligence Research*, vol. 24, pp. 49–79, 2005.

[10] J. Schwartz, R. Zhou, and H. Kurniawati, "Online planning for interactive-pomdps using nested monte carlo tree search," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 8770–8777.

[11] R. Tejwani, Y.-L. Kuo, T. Shu, B. Katz, and A. Barbu, "Social interactions as recursive mdps," in *Conference on Robot Learning*. PMLR, 2022, pp. 949–958.

[12] R. Tejwani, Y.-L. Kuo, T. Shu, B. Stankovits, D. Gutfreund, J. B. Tenenbaum, B. Katz, and A. Barbu, "Incorporating rich social interactions into mdps," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7395–7401.

[13] P. Gmytrasiewicz, "How to do things with words: A bayesian approach," *Journal of Artificial Intelligence Research*, vol. 68, pp. 753–776, 2020.

[14] T. Bolander and M. B. Andersen, "Epistemic planning for single- and multi-agent systems," *Journal of Applied Non-Classical Logics*, vol. 21, no. 1, pp. 9–34, 2011.

[15] C. Muise, V. Belle, P. Felli, S. McIlraith, T. Miller, A. Pearce, and L. Sonenberg, "Planning over multi-agent epistemic states: A classical planning approach," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015.

[16] T. Engesser, T. Bolander, R. Mattmüller, and B. Nebel, "Cooperative epistemic multi-agent planning for implicit coordination," *Electronic Proceedings in Theoretical Computer Science*, vol. 243, 03 2017.

[17] P. H. de Rodrigues Quemel e Assis Santana, S. Thiébaux, and B. C. Williams, "Rao*: An algorithm for chance-constrained pomdp's," in *AAAI Conference on Artificial Intelligence*, 2016.

[18] H. Van Ditmarsch, W. van der Hoek, J. Y. Halpern, and B. Kooi, *Handbook of epistemic logic*. College Publications, 2015.

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

[20] C. L. Baker, J. Jara-Ettinger, R. Saxe, and J. B. Tenenbaum, "Rational quantitative attribution of beliefs, desires and percepts in human mentalizing," *Nature Human Behaviour*, vol. 1, no. 4, p. 0064, 2017.

[21] T. Bolander, "A gentle introduction to epistemic planning: The DEL approach," in *Proceedings of the Ninth Workshop on Methods for Modalities, M4M@ICLA 2017,*, ser. EPTCS, S. Ghosh and R. Ramanujam, Eds., vol. 243, 2017, pp. 1–22.

[22] C. Boutilier, "Planning, learning and coordination in multiagent decision processes," in *Theoretical Aspects of Rationality and Knowledge*, 1996.

[23] A. Baltag and S. Smets, "A qualitative theory of dynamic interactive belief revision," *Logic and the foundations of game and decision theory (LOFT 7)*, vol. 3, pp. 9–58, 2008.

[24] S. Hong and B. C. Williams, "An anytime algorithm for constrained stochastic shortest path problems with deterministic policies," *Artificial Intelligence*, vol. 316, p. 103846, 2023.

[25] J. Lee, G.-H. Kim, P. Poupart, and K.-E. Kim, "Monte-carlo tree search for constrained pomdps," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[26] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," *Advances in neural information processing systems*, vol. 23, 2010.

[27] X. Puig, T. Shu, S. Li, Z. Wang, Y.-H. Liao, J. B. Tenenbaum, S. Fidler, and A. Torralba, "Watch-and-help: A challenge for social perception and human-{ai} collaboration," in *International Conference on Learning Representations*, 2021.