Risk-Bounded Online Team Interventions via Theory of Mind

Yuening Zhang[1], Paul Robertson[2], Tianmin Shu[1], 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 mental states 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 mental states. 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 deadlock. Our experiments and simulated demonstration on the Virtualhome testbed show that the assistant can effectively improve team performance.**

I. Introduction

When a team collaborates on a task, there is often spontaneous coordination on their courses of actions. However, in real life, the team might not have a shared mental model of the task. For example, one may be uncertain of the intent of others or have false beliefs of some task constraints or world state related to 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 of plans to ensure successful execution. Such an assistant can inform members about their false beliefs of the task and of each other, instruct them on what actions to take, and inquire about what beliefs they hold when it 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 mental states and the various forms of communication such an assistant should ideally employ. Meanwhile, dynamic epistemic logic emerged as a useful tool to represent a machine Theory of Mind (ToM) and to model communication in the field of epistemic planning, with applications extended to human-robot teamwork [6].

Consider an example ToM task from [6] in which a robot and a human[3] collaborate to prepare a drink for breakfast. The robot is responsible for grabbing either a mug or a glass as the container, and the human will grab either some coffee or some orange juice for the drink. For the task to succeed, it is required that the mug has to go with the coffee and the glass has to go with the juice. We may consider 4 possible cases related to the team's knowledge of the task constraint as shown in Figure 1. Among those, an external assistant is most useful in the latter two cases. In case (B), a ToM agent as described in [6] taking robot's role can be expected to wait for the human to pick a drink first or explain to the human about the task constraint. However, in (C), if the robot is unaware of the human's false belief and picks a container first, it depends on the assistant to explain the task constraint to the human. Additionally, rather than being certain of which case it is, the assistant may also only have a prior probabilistic belief on which of the 4 cases is true.

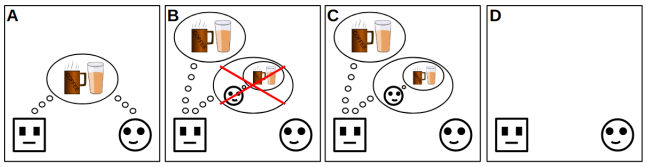

Fig. 1. (A) Robot (square) and human (circle) share knowledge of the task constraint. (B) Robot knows the task constraint and correctly knows that human doesn't know. (C) Robot incorrectly thinks that human also knows the task constraint. (D) Neither agent knows the task constraint.

This work proposes such an AI team assistant assuming it has complete visibility of the team's execution of actions but not their mental states, who intervenes solely through communication, including asking questions, providing explanations, and announcing intent. To prevent overwhelming the team, the assistant is risk-bounded, intervening only to maintain the team's failure rate below a specified risk bound or to resolve execution deadlocks. We show its effectiveness through experiments and demonstration in simulation.

II. Related work

When interaction with another human is concerned, partially observable Markov decision processes (POMDPs) are a common framework in which the human's internal state, such as intent, is hidden [4], [7], [8]. Human behavior is assumed to be conditional on their internal state and stochastic in nature, and their influence on the world is captured by the stochastic transition function. Consequently, these methods typically require pre-specifying or learning a human behavior model pertaining to the transition function, such as learning an AMM [4]. This idea was recently extended to team scenarios [5], proposing an AI agent that provides task-time interventions. Their method, however, solely instructs teams on what intent to pursue, since 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

[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]Agents are equal partners and the role of human/robot does not matter.

recursive belief representation – a explicit model of Theory of Mind, such as interactive POMDP (I-POMDP) [9], [10] or social MDP [11], [12]. Each agent can predict others' actions by solving a nested problem according to its belief of their model, assuming agents are mostly rational and reward-maximizing. Since agents' beliefs have a direct impact on their actions, this has the benefit that we can influence their actions by altering their beliefs. However, these frameworks have limited applicability in practice due to their intractability, the need for a complex model for explicit communication that directly alters beliefs [13], and it is impractical to assume accurate prior data on such nested finite probabilistic spaces.

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, the DEL approach [14], [16] leverages dynamic epistemic logic (DEL) to model the update of the epistemic state due to both physical and epistemic actions. Recent work, EPike [6], shows the applicability of the DEL approach to human-robot teamwork. Our method is an innovative combination of insights from EPike and the POMDP formulation to produce a risk-bounded assistant with a explicit Theory of Mind.

III. Problem Formulation

We formulate the assistant's intervention problem based on the POMDP framework, where the hidden state is the team's epistemic state, i.e. the players' nested mental models of the task, represented using epistemic logic. We leverage two key ideas: (1) state transitions, especially as a result of communication actions, can be defined using dynamic epistemic logic (DEL), and (2) the observations consist of actions taken by the team, whose probability can be estimated using epistemic planning techniques given a specific epistemic state and assuming agents are mostly rational, which in turn reveal the true epistemic state, similar to inverse planning [17], [18].

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 subset 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 possible intervention actions $\mathcal{A}_I$ and possible actions by 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 function 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) We introduce a safety function, 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 an agent in the team takes an action that leads to failure of the task. (2) Besides the assistant taking intervention actions, an observation, in this case, is also an action taken by the team and provides an additional state transition, incurring further cost and potentially resulting in failure as well. By employing the safety function, our formulation builds on the chance-constrained POMDPs (CC-POMDPs) [19] and requires an additional specification of a risk bound Δ, the maximum allowed probability of failure. 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:

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

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 occurs in the state trajectory.

A. Calculating Risk of Failure

To ensure the satisfaction of the chance constraint Eq (2), we introduce $er(b_t|\pi)$ that represents the execution risk of policy π from belief state b_t for the remainder of the planning horizon. Eq (2) then amounts to satisfying the condition $er(b_0|\pi) \leq \Delta$. In order to calculate the execution risk, we follow CC-POMDPs [19] and count the probability of a transition into a failure state towards the risk spent, and we continue acting conditional on no failure having occurred. We further define safe belief $\tilde{b}_t$ as belief at time t conditional on all states being safe, that is, the leading transitions to the state all satisfy the safety function δ. The initial belief is assumed to be safe: $\tilde{b}_0 = b_0$.

For a single step s_t, given safe belief $\tilde{b}_t$ and intervention action a, the updated prior belief b'_{t+1} for s'_{t+1} is:

$$b'_{t+1}(s') = \sum_{s} \delta(s, a) T(s, a, s') \tilde{b}_t(s) \quad (3)$$

with the probability that state s'_{t+1} is safe from intervention a denoted as $p_{sa}(\tilde{b}_t, a) = \sum_s \delta(s, a)\tilde{b}_t(s)$. The safe belief $\tilde{b}'_{t+1}$ can then be obtained by normalizing over the probability that it is still safe:

$$\tilde{b}'_{t+1}(s') = b'_{t+1}(s')/p_{sa}(\tilde{b}_t, a) \tag{4}$$

Given $\tilde{b}'_{t+1}$, the probability of observing $o_{t+1} = o$ is:

$$Pr(o_{t+1} = o|\tilde{b}'_{t+1}) = \sum_{s'} \tilde{b}'_{t+1}(s')O(s', o) \tag{5}$$

Assuming that $o_{t+1} = o$ is observed, we can first update our posterior belief of $\tilde{b}'_{t+1}$:

$$\tilde{b}'_{t+1}(s'|o) = \frac{1}{\eta} O(s', o)\tilde{b}'_{t+1}(s') \tag{6}$$

where $\eta = \sum_{s'} O(s', o)\tilde{b}'_{t+1}(s')$. Since observation provides another state transition, we can compute the posterior belief after the transition as:

$$b_{t+1}(s) = \sum_{s'} \delta(s', o)T(s', o, s)\tilde{b}'_{t+1}(s'|o) \tag{7}$$

with the probability that state s_{t+1} is safe from observation o denoted as $p_{sa}(\tilde{b}'_{t+1}, o) = \sum_{s'} \delta(s', o)\tilde{b}'_{t+1}(s'|o)$. The safe belief $\tilde{b}_{t+1}$ conditional on the observation being safe is:

$$\tilde{b}_{t+1}(s) = b_{t+1}(s)/p_{sa}(\tilde{b}'_{t+1}, o) \tag{8}$$

Given the above, we can compute the execution risk from any safe belief $\tilde{b}_t$ by evaluating the risk spent at each step s_t and the subsequent cumulative risk assuming that s_t hasn't failed. For simplicity, we focus on deterministic policies here while stochastic policies can be derived similarly:

$$er(\tilde{b}_t|\pi) = 1 - p_{sa}(\tilde{b}_t, a) + p_{sa}(\tilde{b}_t, a)er(\tilde{b}'_{t+1}|\pi) \tag{9}$$

where, denoting $Pr(o_{t+1} = o|\tilde{b}'_{t+1})$ as $p_{obs}(\tilde{b}'_{t+1}, o)$,

$$\begin{aligned} er(\tilde{b}'_{t+1}|\pi) = \sum_o p_{obs}(\tilde{b}'_{t+1}, o)\big(1 - p_{sa}(\tilde{b}'_{t+1}, o) + \\ p_{sa}(\tilde{b}'_{t+1}, o)er(\tilde{b}_{t+1}|\pi)\big) \end{aligned} \tag{10}$$

Similarly, the expected cost $c(\tilde{b}_0|\pi)$ for a given policy π can be computed recursively from: $c(\tilde{b}_t|\pi) = \sum_s C(s, a)\ \tilde{b}_t(s) + p_{sa}(\tilde{b}_t, a)c(\tilde{b}'_{t+1}|\pi)$, where $c(\tilde{b}'_{t+1}|\pi) = \sum_o p_{obs}(\tilde{b}'_{t+1}, o)\ \ (\sum_{s'} C_O(s', o)\tilde{b}'_{t+1}(s'|o) + p_{sa}(\tilde{b}'_{t+1}, o)\ c(\tilde{b}_{t+1}|\pi))$.

B. 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 to intervene or not and to intervene consecutively if needed, such as asking a question to one player 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 $C_O(s, skip) = 0$.

C. State Transitions via DEL

The primary advantage of DEL-POMDP is its utilization of DEL for defining the transitions of the team's mental state in response to both physical and epistemic actions. In this section, we describe the general framework pertinent to any DEL formulation. The subsequent section will delve into our specific choice of DEL for multi-agent task execution.

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 [20], which encompasses not only the actual state but also agents' informational perspectives of the state. We require each epistemic state to be *global*, that is, the pointed Kripke model is pointed at exactly one world, or intuitively speaking, each state fully determines the information attitude of the agents. The assistant's uncertainties are captured by its belief state – a probability distribution over global epistemic states. We refer to the belief over global epistemic states as an *epistemic belief state*.

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 [21] with a similar Kripke structure. An action a updates the 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 at $s \in \mathcal{S}$, the set of possible interventions at an epistemic 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-D.2.

Note that in DEL, agents are assumed to execute asynchronously instead of taking joint actions simultaneously as in typical multi-agent MDPs [**?**]. We assume the availability of such an observation function O that outputs the list of possible team actions and their probabilities $O(s, o)$ given an epistemic state s. However, such O may not be readily available. In our case, it is more reasonable to assume the availability of function O_i that outputs the probability of actions for each agent i in the team, estimated using existing techniques in epistemic planning [6], [16], and compute O from O_i. More specifically, when considering the team's 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 hanging execution and no progress being made. Denoting inaction as $noop$, the team only takes a $noop$ if none of the agents act. Thus, the likelihood of a team-wise $noop$ at state s is computed as $O(s, noop) = \prod_{i \in Ag} O_i(s, noop)$. We then normalize the probability for any non-noop action for agent i by: $O(s, o_i) = \frac{O_i(s, o_i)}{\sum_{j \in Ag} \sum_{o_j \neq noop} O_j(s, o_j)}(1 - O(s, noop))$.

Finally, we define two task-specific concepts. First, we define a set of *successful* global epistemic states G, where $s \in G$ denotes a state that reaches the end goal of the task.

Such a goal may be the team having successfully prepared the drink, or it can also include epistemic goals, such as when all the team members know that they have reached the goal state, or even that there is common knowledge that they have reached the goal state. Second, we define a set of *failure* global 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. We define our success and failure state for multi-agent task execution in Section III-D.3.

Given the above, we provide the following specification to the DEL-POMDP functions:

- $\delta(s, a) = 1$ if and only if action a is applicable to s and $s' = s \otimes a \notin F$.
- $T(s, a, s') = 1$ if and only if a is applicable to 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 to 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 function allows the assistant to take some risks, such as when the likelihood of the team taking an incorrect action is small, or when an intervention only has a small likelihood of being inapplicable, such as making an announcement of something that the assistant almost certainly believes to be true.

D. DEL for Multi-Agent Task Execution

To represent 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 is defined by plausibility models [22] instead of Kripke models. At a high level, such a logic differs from the most commonly seen epistemic logic in 2 ways: (1) first, 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, i.e. what plans are feasible. (2) Second, *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 of an assistant in the framework, while referring the readers to [6] for the details of CDL-KB.

1) Epistemic Belief State: The assistant holds an epistemic belief state – a probability distribution over the set of epistemic states capturing possible team mental states on the task. Formally, a global epistemic state is represented by a pointed plausibility model $s = (M, w)$, where M is a plausibility model that specifies the information perspectives of the agents, and $w \in W$ is a pointed world representing the actual plan space for the task. Note that the set of agents Ag defined for M does not include our assistant, meaning that agents in the team do not actively consider the assistant's participation in the task. We emphasize again that while the assistant's belief is probabilistic, epistemic states represented by plausibility models are qualitative and do not contain probabilities. Figure 2 shows the corresponding modeling of epistemic states from Figure 1.

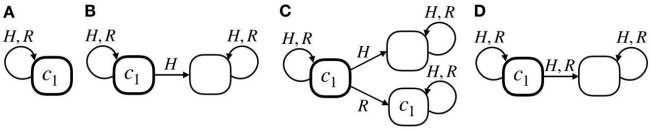

Fig. 2. Epistemic states with human (H) and robot (R), where each node is a possible world representing a possible knowledge base for the task, with the pointed world highlighted in bold. The arrows represent the order of plausibility from the agents' perpsectives. c_1 represents the task constraint.

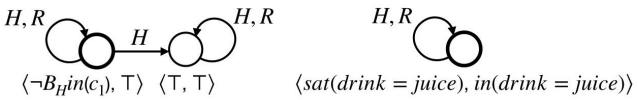

Fig. 3. (left) The assistant's explanation is modeled by the precondition $\neg B_H in(c_1)$, i.e. human does not believe c_1 is in the knowledge base. The arrow to the right indicates that human thought nothing happened, hence the explanation is private to the robot. (right) Announcing intent to be juice requires the precondition that $sat(drink = juice)$, i.e. selecting juice is satisfiable, and adds the constraint $(drink = juice)$ as an effect.

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

Explanation & Intent Announcement: An explanation action allows an agent to explain a task constraint or someone's belief to another who is uncertain or has incorrect belief. For agent i to explain φ, the precondition is that the agent itself must believe what it explains is true, i.e. $B_i\varphi$. An intent announcement allows an agent to commit the team to specific choices of actions. For agent i to announce the intent c, the precondition is that agent i believes the intent is satisfiable, i.e. $B_i sat(c)$. The precondition, conversely, also dictates 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 considering these actions coming from the assistant, we assume the assistant's words are always taken as the truth by the agents, and simply remove any prefix of B_i from the precondition. Figure 3 shows two example interventions where the assistant privately explains to the robot that the human does not know about the task constraint (left), and the assistant announces the intent for the human to select orange juice (right).

Asking Questions: Agents can ask each other questions to inquire about their beliefs concerning φ. For example, one might ask another agent if they know the task constraint or not. In DEL, this is modeled as a non-deterministic action. When the assistant asks a question, we can model the effect of such a 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

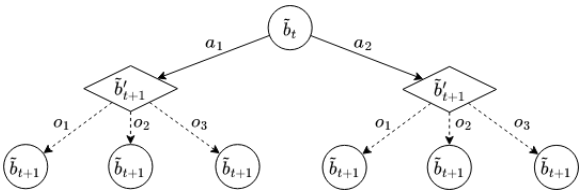

Fig. 4. Single-step expansion for the DEL-POMDP explicit graph

answer to the question being announced privately, modeled as a private announcement action. $O(s, o) = 1$ iff $s \vDash pre(o)$, where pre denotes precondition. The probability of receiving each observation is computed based on Eq (5).

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.

Note that with this formulation, an intervention will not result in failure as long as it is applicable to s. Additionally, because the precondition for an agent's action depends only on the agent's subjective belief, an action is applicable to s from the agent's local perspective if and only if it is actually applicable to s. This means while an inapplicable intervention to s results in failure, an observed action inapplicable to s, instead, provides evidence that s is not the true state.

E. Estimating Probability of Observations

To provide an estimation of O_i for each agent i, we assume agents are bounded rational and will not take an action that they consider inapplicable. We take advantage of the online execution planning algorithm from EPike that outputs a list of applicable actions for an agent and their subjective scores from the agent's perspective at a given epistemic state s [6]. We assume an agent has a uniform probability of taking any action from the set that maximizes the subjective score. As an MCTS-based algorithm, we can control how rational the agents are by varying the number of iterations to run for.

IV. Risk-Bounded Algorithms

To find a policy π that meets the risk bound Δ, we first illustrate the policy space by expanding an explicit graph up to horizon h with each step of the expansion shown in Figure 4. A policy is a subtree of this explicit graph. For deterministic policies, a unique action is chosen at every circle node, whereas for stochastic policies, we determine a probability distribution of the actions at every circle node [23]. We use three state-of-the-art risk-bounded algorithms to solve for the policy.

RAO* [19] 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 introduce 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 cumulative cost at each node in Figure 4 as defined in Section III-A.

ACDC [24] is an anytime optimal solver for constrained POMDPs (CPOMDPs) producing deterministic policies. It consists of two phases, where 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 4 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, producing stochastic policies and converging to the optimal policy over time. To handle the constraints, it uses the dual formulation to convert the problem into an unconstrained one while optimizing 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 III-A.

For our assistant to execute online, there are two options. In the case that the system is guaranteed to terminate within h steps, where h is the horizon, we 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 only the first action of the policy is taken at each step, and we replan for a new policy 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 system keeps track of the probability that it is still safe P_{sa}. Every time an intervention action a_t is taken and an observation o_{t+1} is received, the belief is updated according to Eq (4) and (8), with P_{sa} multiplied by the probability of safety $p_{sa}(\tilde{b}_t, a)$ and $p_{sa}(\tilde{b}'_{t+1}, o)$. The system 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 multi-agent task execution 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, we run EPike's MCTS algorithm

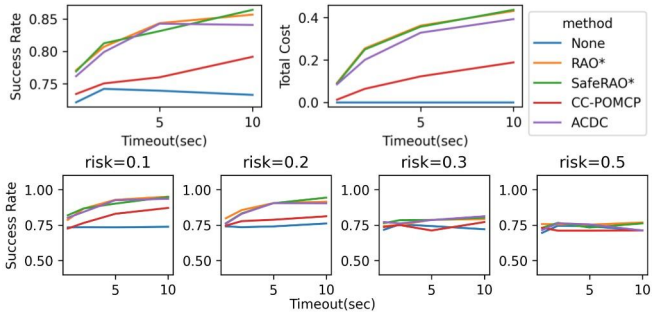

Fig. 5. Success rate and total cost with and without assistant

for 100 iterations. For observations of team actions, we sample from our computed estimation of probability.

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 execute actions, 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. We sample from 1 to 5 actions per agent. Additionally, we sample from 0 to 3 task constraints that the agents are missing collectively, i.e. some agent does not have knowledge of the constraint. For each missing constraint, a probabilistic distribution of two out of four cases in Figure 2 is assigned.

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. Figure 5 top row shows the averaged result for the generated testcases over all the risk bounds for the different risk-bounded algorithms. The results show that the assistant significantly increases the success rate given enough time, with (safe) RAO* being the most efficient at finding a solution. This is likely because RAO* requires fewer calls to estimate the probability of observations due to its pruning of the policy space, where the majority of the computation time is spent. The bottom row shows the success rate under different risk bounds. We see that the success rate decreases as we increase the risk bound, especially for CC-POMCP, since it employs stochastic policies that can maximize its utilization of risk.

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 with a risk bound of 0.05. First, we consider the **Breakfast** domain as described in Figure 1. We integrated the EPike algorithm [6] and our assistant algorithm with the Virtualhome simulator [27], and demonstrate 3 scenarios detailed in the attached video. Second, we consider a **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

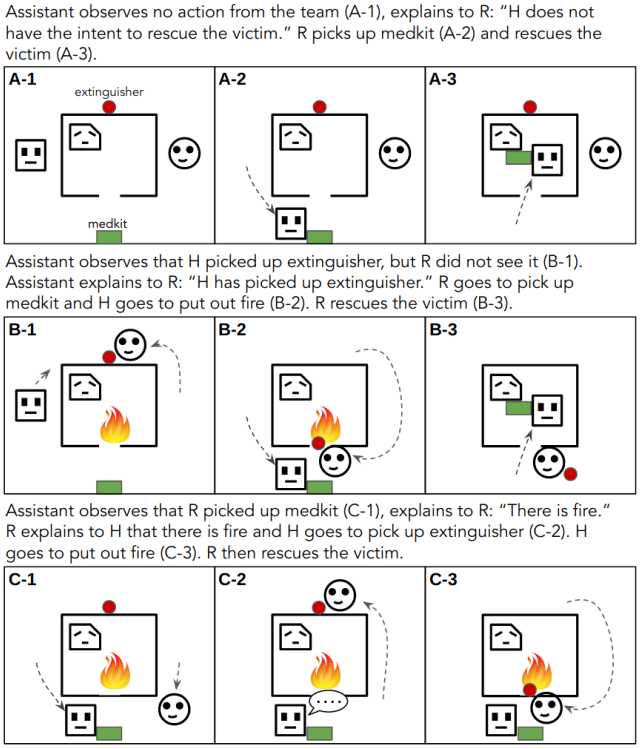

Fig. 6. Scenarios: (A) There is some probability that both agents think the other agent will rescue the victim. (B) Both agents know about the fire. (C) There is some probability that neither agent knows about the fire.

each agent. We run the algorithm on 3 scenarios and illustrate the results in Figure 6. 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 risk-bounded assistant has the benefit that: (1) It assumes a higher priority than regular agents in the team. (2) It considers the probabilities of potential epistemic states, instead of treating them with equal weights. (3) Its belief state is updated via Bayesian posterior upon observation of agents' actions whereas an agent's belief is only revised through DEL action update. However, our work has the limitations that it does not handle contingent world state observations and requires the initial pool of possible epistemic states to be provided. Future work could incorporate agents' beliefs of the current world state and consider contingent plans [21], and 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 assistant that improves teamwork through interventions. We validated through experiments and simulation that the assistant is effective in improving team performance.

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

[18] 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.

[19] 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.

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

[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] 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.

[23] E. Altman, *Constrained Markov decision processes*. Routledge, 2021.

[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.