Reasoning and Explanation Generation in Ad hoc Collaboration between Humans and Embodied AI

Hasra Dodampegama[0000−0003−2302−1501] and Mohan Sridharan[0000−0001−9922−8969]

Institute of Perception, Action and Behavior, School of Informatics, University of Edinburgh,UK
{hasra.dodampegama, m.sridharan}@ed.ac.uk

Abstract. An assistive embodied AI agent often has to collaborate with previously unseen humans. State of the art frameworks for such *ad hoc teamwork* use a large labeled dataset of prior observations to model the behavior of other agents and to determine the ad hoc (i.e., embodied AI) agent's behavior. These approaches do not support rapid incremental revisions or transparency, and the necessary resources (e.g., training examples, computation) are not readily available in practical domains. Our previous work introduced an architecture that enabled an ad hoc agent to choose its actions in simple simulated domains based on non-monotonic logical reasoning with prior domain knowledge and models learned from limited examples to predict the behavior of other agents. Here, we extend this architecture to enable an ad hoc embodied AI agent to collaborate with a human performing household tasks in a complex indoor environment, focusing on the ad hoc agent's ability to identify and reason with relevant knowledge, and provide relational descriptions as explanations of its behavior and that of the human. We evaluate our architecture's capabilities in *VirtualHome*, a realistic 3D simulation environment.

Keywords: Ad hoc teamwork · Non-monotonic logical reasoning · Ecological rationality · Explanation generation · Embodied AI.

1 Introduction

The screenshots in Figure 1 show an embodied AI agent collaborating with a previously unseen human to perform household tasks (e.g., make breakfast, clean dishes) in a multiagent simulation environment. The agent has to reason with different descriptions of some commonsense domain knowledge and uncertainty. This includes logic-based and metric descriptions of some attributes of the domain and the agent, some rules governing actions and change, and default statements that are true in all but a few exceptional circumstances. The agent may have to revise its knowledge and action choices in response to domain changes. Also, the agent has a limited view of the environment and limited communication bandwidth. This scenario is an instance of Ad Hoc Teamwork (AHT), the problem of enabling an agent to cooperate with others without prior coordination [22], which arises in many practical applications.

The state of the art in AHT has moved from using predetermined policies for selecting actions in specific states to methods with a key *data-driven* component that uses a long history of prior experiences to build probabilistic or deep

Fig. 1. Screenshots from the *VirtualHome* environment [16], showing a human (female in green top) and an embodied AI agent (male in blue shirt) collaborating.

network methods that model the behavior of other agents (or agent types) and optimize the behavior of the ad hoc agent [15]. However, it is difficult to gather large training datasets of different situations in practical domains. Also, these methods lack transparency, and make it difficult to revise existing knowledge over time. Unlike existing work, our prior work developed a knowledge-guided architecture for AHT, enabling an ad hoc agent to determine its actions based on non-monotonic logical reasoning with prior domain knowledge and learned predictive models of other agents' behaviors [8, 9]. In this paper, we describe an extension, REACT, which considers a more complex embodied environment for human-agent collaboration, focusing on two key capabilities:

1. Automatically identify and perform non-monotonic logical reasoning with relevant commonsense domain knowledge and a rapidly-learned predictive model of a human agent to determine an ad hoc agent's actions; and
2. Introduce and implement a methodology to automatically construct on-demand relational descriptions of the decisions of the ad hoc agent and the human as *explanations* in response to different types of questions.

We evaluate these capabilities in household scenarios in *VirtualHome*, a realistic physics-based 3D simulation environment for multiagent collaboration [16]. We first describe related work (Section 2), followed by our architecture (Section 3), results of experimental evaluation (Section 4), and conclusions (Section 5).

2 Related Work

As discussed in a recent survey on AHT [15], early work used specific protocols to define the agents' behavior in different scenarios [6] while subsequent work used sampling methods [5]. State of the art methods include a key data-driven component, using probabilistic, deep-network and/or reinforcement learning methods to learn action choice policies for different types of agents based on a long history of prior observations of similar agents or situations. For example, attention-based deep neural networks have been used to jointly learn policies for different agent types [7] and account for different team compositions [17], and sampling strategies have been combined with learning methods to optimize performance [23]. Such methods require considerable computation, memory, and training examples, build opaque models, and make it difficult to adapt quickly to changes.

There is considerable research in action languages and logics for multiagent domains. This includes action language $\mathcal{A}$ for an agent computing cooperative actions in multiagent domains [19], and recent work on action language m$\mathcal{A}*$ that introduces action types, epistemic planning, and dynamic awareness to model realistic interactions [4]. Our prior AHT architecture translated an action language description of prior knowledge to a program in Answer Set Prolog (ASP), a non-monotonic logical reasoning paradigm. For any given goal, the ad hoc agent computed a plan by reasoning with this knowledge and the predicted action choices of other agents (obtained from learned models) [9].

Embodied AI refers to AI systems operating within physically realistic (simulation) environments such as Habitat [18] and VirtualHome [16]. These interactive platforms support the generation of complex scenarios for evaluating algorithms for single-agent and multi-agent collaboration problems. With the increasing use of embodied AI agents and AI methods in different applications, there is renewed focus on transparency and explainability of decision-making [1]. Work in our group on a theory of explanation generation has been implemented within a refinement-based architecture (that uses ASP for reasoning) for robots [21].

The architecture described in this paper builds on and extends our prior work in two significant ways: (i) automatically identifies and reasons with relevant knowledge, reducing the search space to support scaling to complex domains; and (ii) provides a methodology for generating relational descriptions as explanations in response to different kinds of questions.

3 Architecture

Figure 2 is an overview of our architecture, *Reasoning and Explanations for Ad hoc Collaboration in Teams* (REACT), for human-embodied AI collaboration. The embodied AI agent is the ad hoc agent that performs non-monotonic logical reasoning with prior commonsense domain knowledge and an incrementally learned behavior model of its teammate (i.e., human). At each step, each teammate receives state observations, and independently determines and executes an action. REACT's components are described using the following example.

Example Domain 1 *[Example Embodied AI Agent Domain]*
Consider an embodied AI agent and a human collaborating to complete household tasks; Figure 1 shows snapshots while preparing breakfast [16]. The agent and the human can interact with the environment through high-level actions, e.g., move to places, pick up or place objects, switch appliances on or off, and open or close appliances. Completing a task requires a sequence of such actions to be computed and executed by the embodied AI agent or the human who do not communicate directly with each other. Also, the agent assumes that the human has access to the same state information and will make (what it considers as) rational decisions. Prior commonsense knowledge of the embodied agent includes relational descriptions of some attributes of the domain, ad hoc agent, and human (Section 3.1); a learned (or encoded) graph of information about likely locations of objects in the domain; and default statements that hold in all

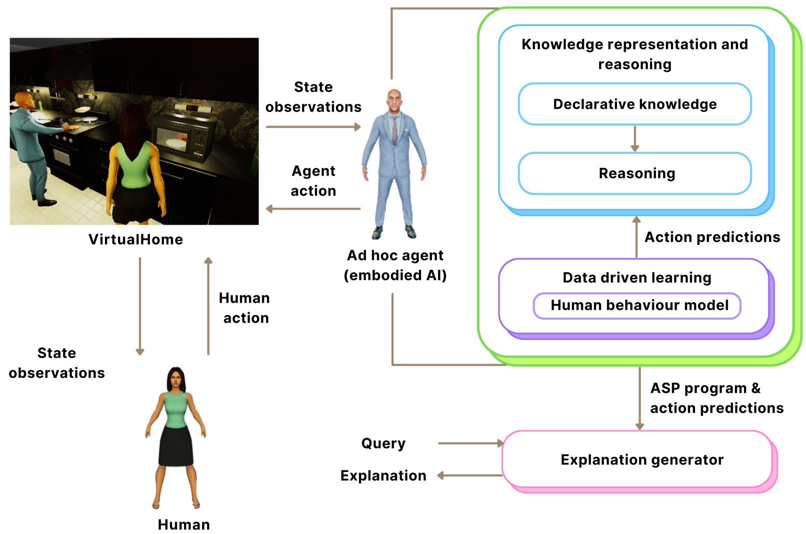

Fig. 2. REACT architecture

but a few exceptional circumstances. The knowledge also includes some axioms governing actions and change, e.g., the agent or human can only pick one object at a time, and certain food items require preparation prior to consumption.

3.1 Knowledge Representation and Reasoning

In REACT, the transition diagram of any given domain is described using an extension of action language $\mathcal{AL}_d$ [12]. REACT's domain representation comprises a system description $\mathcal{D}$, a collection of statements of $\mathcal{AL}_d$, and a history $\mathcal{H}$. $\mathcal{D}$ has a sorted signature Σ with basic sorts such as *object*, *appliances*, *ad_hoc_agent*, *human*, and *step* (for temporal reasoning) in our example domain; actions such as $grab(ad\_hoc\_agent, object)$ and $switch\_on(ad\_hoc\_agent, appliances)$; statics, i.e., domain attributes whose values cannot be changed by actions; and fluents, i.e., attributes whose values can be changed by actions. Basic sorts (and actions) are arranged hierarchically, e.g., *microwave* is a sub-sort of *electricals* that is a sub-sort of *appliances*, a sub-sort of *objects*; and the action *find* can include a series of *move* and *rotate* actions. Fluents can be *inertial*, i.e., they obey laws of inertia and are changed by actions, e.g., $in\_hand(ad\_hoc\_agent, object)$ describes an object being held by the ad hoc agent; and *defined*, i.e., they do not obey inertia laws and are not directly changed by the ad hoc agent's actions, e.g., $agent\_hand(human, object)$ describes the human holding an object.

Based on Σ, the domain dynamics are described in $\mathcal{D}$ using three types of axioms: *causal law*, *state constraint*, and *executablility condition* such as:

$$grab(A, O) \textbf{ causes } in\_hand(A, O) \tag{1a}$$

$$heated(F) \textbf{ if } on(F, E), switched\_on(E) \tag{1b}$$

$$\textbf{impossible } grab(A, O) \textbf{ if } on(O, E), not\ opened(E) \tag{1c}$$

where Statement 1(a), a causal law, implies that grabbing an object causes it to be in the hand of the ad hoc agent; Statement 1(b), a state constraint, implies that a food item placed in an electrical appliance (e.g., microwave) that is switched on gets heated; and Statement 1(c), an executability condition, prevents the ad hoc agent from trying to grab an object from an appliance with a closed door. History $\mathcal{H}$ is a record of observations of the form $obs(fluent, boolean, step)$, and action executions of the form $hpd(action, step)$ at specific time steps. It also includes default statements that are true in the initial state.

To reason with knowledge, we automatically construct program $\Pi(\mathcal{D}, \mathcal{H})$ in CR-Prolog [3], an extension to ASP that supports consistency restoring (CR) rules. $\Pi(\mathcal{D}, \mathcal{H})$ includes statements from $\mathcal{D}$ and $\mathcal{H}$, inertia axioms, reality check axioms, closed world assumptions for defined fluents and actions, helper relations, e.g., $holds(fluent, step)$ and $occurs(action, step)$ to imply that a fluent is true and an action is part of a plan at a time step, and helper axioms that define goals and guide planning and diagnosis. ASP encodes *default negation* and *epistemic disjunction*, and supports non-monotonic logic reasoning. This ability to revise previously held conclusions is essential for agents reasoning and acting in practical domains based on incomplete knowledge and noisy observations. The CR rules allow the agent to make assumptions (e.g., that a default statement does not hold) under exceptional circumstances to recover from inconsistencies. All reasoning tasks (i.e., planning, diagnostics, and inference) are then reduced to computing *answer sets* of Π. We use the SPARC system [2] to solve CR-Prolog programs. Example programs and results are in our open source repository [10].

Our current application domain is substantially more complex than those in our prior work, with many more objects and actions; the corresponding domain description is more complex and the tasks require much longer plans. To enable the use of the KR formalism (above), REACT incorporates new strategies to constrain the search space. Specifically, as the ad hoc agent traverses through our household environment or in other similar environments, it can collect statistics from observations, e.g., of relevant locations of objects, human action preferences. Any such (learned or encoded) knowledge is used to automatically restrict the grounding and simplify the processing. For example, depending on the goal and specific actions in the plan, the ad hoc agent can automatically select the relevant signature and restrict the axioms to this reduced signature.

3.2 Agent Behavior Models

The human's actions also revise the domain state, which determines the ad hoc agent's action choices. REACT thus reasons with prior knowledge and models that predict the behavior of the human. Our prior work introduced the use of the *Ecological Rationality* (ER) principle [13], which is based on Herb Simon's definition of *Bounded Rationality* and the algorithmic theory of heuristics, to rapidly learn and revise these predictive models. The ER-based approach enables the ad hoc agent to choose relevant attributes and learn models of the human's behavior from limited data while supporting rapid, incremental updates.

Specifically, REACT enables the ad hoc agent to learn an ensemble of *Fast and Frugal* (FF) trees that predict the behavior of the human; each FF tree pro-

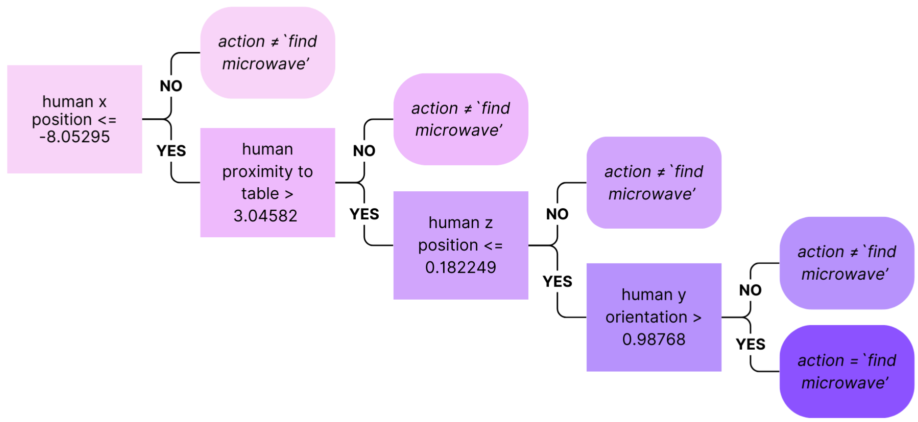

Fig. 3. FF tree model of human behavior for the *find(microwave)* action.

Table 1. Attributes used to create behavior models of human.

Description of the attribute
Immediate two previous actions of the human
Immediate two previous objects human interacted
Position of the human (x,y,z)
Orientation of the human (x,y,z)
Distance from human to the kitchen table
Distance from human to the kitchen counter
Number of objects on the kitchen table

vides a binary choice for a particular action and the number of leaves in the tree is limited by the number of attributes [14]. One FF tree in an ensemble learned for the human is shown in Figure 3. The initial version of these trees were built using only 100 traces of human action choices and domain state from the Virtual-Home domain, with the corresponding attributes listed in Table 1. Furthermore, consistent agreement (disagreement) between observations and predictions of an existing model triggers model choice (revision); the ad hoc agent is thus able to quickly adapt to changes in the domain or the other agent.

3.3 Transparency in decision making

An automated decision-making system's ability to answer questions about its decisions promotes acceptability [1, 11]; this transparency also plays an important role in human reasoning and learning. Unlike existing methods that seek to make a complex learned model interpretable, or to explain (or justify) all the choices made by a reasoning system, REACT seeks to respond to any given question about specific decisions made by the ad hoc agent or the human by quickly identifying the relevant information and constructing relational descriptions.

REACT's use of knowledge-based reasoning and simple predictive models is the foundation for the approach introduced in this paper to provide the desired on-demand descriptions. We build on prior work in our group that demonstrated the ability to identify the axioms and literals relevant to the questions posed to a system making automated decisions [20]. The agent generates relational descriptions in response to four types of questions identified as being important in work on explainable planning [11]. Here we describe the procedure the ad hoc agent follows to generate responses for these types of questions.

1. **(Causal questions)** *Why did you execute a_I, i.e., action a at step I?*
   - If a_I is not the last action of plan P executed by the agent, extract actions $\{A_{af}\} \in P$ that occurred immediately after $a_I \in P$.
   - Examine $\Pi(\mathcal{D}, \mathcal{H})$ to identify axioms with the negation of $a_{I+i} \in \{A_{af}\}$ in *head*, i.e., axioms encoding conditions that prevent a_{I+i} from occurring.
   - Check if each such axiom's *body* is satisfied by *answer set* at time step I. If yes, identify fluent literals f in *body* whose value changed from I to $I+1$.
   - All such literals $\{f\}$ over all identified axioms have been changed by the execution of a_I. Collect these literals to construct the answer.
   - If a_I is the last action in P, it contributed directly towards achieving the goal. Use goal G and a_I to construct the answer.
2. **(Contrastive questions)** *Why did you not execute a_I, i.e., action a at I?*
   - Examine $\Pi(\mathcal{D}, \mathcal{H})$ to identify axioms with the negation of a_I in its *head*, i.e., executability conditions.
   - Check if each selected axiom's *body* is satisfied by *answer set* at I. If yes, collect fluent literals $\{f\}$ in *body* as they prevented consideration of a_I.
   - If not, examine $\Pi(\mathcal{D}, \mathcal{H})$ to identify axioms with a_I in its *body* alongside other literals $\{f'\}$, i.e., causal laws.
   - Extract the additional precondition literals $\{f'\}$. Examine $\Pi(\mathcal{D}, \mathcal{H})$ to identify axioms with $l \in \{f'\}$ in its *head*, i.e., state constraints.
   - Check if selected axiom's *body* is satisfied by *answer set* at step I. If not, use literal $l \in \{f'\}$ and the *body* literals of axiom to construct answer.
3. **(Justify beliefs)** *Why did you believe l_I, i.e., l at step I?*
   - Replace the grounded terms of l_I with the corresponding variables.
   - Examine $\Pi(\mathcal{D}, \mathcal{H})$ to identify axioms with l_I in *head*. i.e., state constraints.
   - Check whether each selected axiom's *body* is satisfied by the answer set at step I. If yes, collect fluent literals $\{f\}$ in *body* as they support belief l_I.
   - If multiple axioms are identified, select one of them to provide explanation.
   - If belief tracing is enabled, create a tree with l_I as its head and each selected axiom as a branch. With the axiom, also store the supporting $\{f\}$ fluents.
   - Repeat procedure for each fluent literal in $\{f\}$ as target belief until no more axioms are identified. An example of a belief tree is in Figure 4.
4. **(Counterfactual Questions)** *What will be the outcome if you (or human R) execute a'_I in (future) step I? What will be the resultant world state at (future) step I if the human executes actions predicted by learned models?*
   - Retrieve most recent state observation of environment S_{I-n} in relation to step I. i.e., start with the current best estimate of world state.
   - Use the predictive behavior models of human(s) to retrieve their action $\{A^{ot}_{I-n}\}$ for step $I-n$. REACT provides action for the ad hoc agent a_{I-n}.

- Perform mental simulation of the future step $I-n+1$ from S_{I-n} using existing knowledge and action choices of human(s) $\{A^{ot}_{I-n}, a_{I-n}\}$.
- Repeat above n times, i.e., roll out the future and explore effects of the action of ad hoc agent and human until environment reaches the queried step I. Collect the state information S_I at I.
- Feed S_I to the predictive behavior model of the human R (REACT) to retrieve the action a^o_I (a_I) for R (ad hoc agent) in state S_I.
- Traverse through the FF tree model of R to identify active branches when selecting a^o_I. Collect a^o_I and these branches to construct the answer.
- Replace action a^o_I (a_I) with a'_I for the human R (ad hoc agent).
- Roll out environment one step to obtain the resultant state S_{I+1}. Collect state information S_{I+1} to construct the explanation.

The acquired information may be used for further training, especially the human behavior prediction models. For all types of questions, the identified literals are processed using existing tools and templates to generate textual descriptions provided as responses (i.e., explanations) before, during, or after planning or execution. Section 4.3 provides execution examples.

4 Experimental Setup and Results

We experimentally evaluate two hypotheses regarding the capabilities of REACT in the VirtualHome domain:

- **H1:** The combination of reasoning and learning in REACT enables the ad hoc agent to adapt to changes and perform better than a logic-based baseline;
- **H2:** REACT enables the ad hoc agent to generate relational descriptions as explanations of its decisions and beliefs and those of the other agents.

The performance measures were the number of steps (i.e., plan length) and the total time taken by the team (i.e., human and embodied AI agent) to complete the task. Details of experiments and baselines are provided below.

4.1 Experimental Setup

In the VirtualHome domain, we modeled the human as a simulated entity that chooses its actions based on an ASP program. The human was assigned the same goal as the embodied AI (ad hoc) agent, e.g., *prepare and eat breakfast*. The **baseline operation** involved each of them receiving the same observations of the domain at each step; they then continued with their planned actions or computed a new plan as needed. There was no direct communication between them, and they did not have any prior knowledge or model of each other's capabilities. With REACT, the ASP program for the ad hoc agent and that for the human were similar. The key difference was that the ad hoc agent used the learned models to predict a couple of actions of the human, and its ASP program included additional axioms for reasoning about these predicted actions. *The agent's plan thus expects the preconditions for some intermediate steps to be created by the actions executed by the human although the human may not do*

Table 2. Mean and standard deviation of no. of steps and time taken by REACT architecture to achieve the goal, expressed as a fraction of these values for the baseline.

Architecture	**Steps**	**Time**
REACT	0.89 ± 0.11	0.90 ± 0.19
Baseline	1 ± 0.05	1 ± 0.04

so. The human, on the other hand, could not predict the ad hoc agent's actions and the axioms to reason about these actions were not included in the corresponding ASP program. The human's actions were primarily determined by the current state. The human's ASP program did include actions such as eating and drinking that were not available to the ad hoc agent. Furthermore, the human's ASP program encoded priorities and preferences, e.g., when preparing breakfast, the human toasted the bread first before preparing the cereal.

As described in Section 3.2, each predictive model was an ensemble of FF trees based on just 100 prior traces of human actions and domain state. These observations also provided priors regarding the likely locations of objects; this information is used to simplify planning (end of Section 3.1).

To evaluate **H1**, we designed **Exp1** by first constructing 720 different configurations, each with a different arrangement and status of objects in the initial condition, e.g., bread on the kitchen table instead of the counter, microwave open instead of closed. We then randomly chose 100 of these configurations and measured the ability to achieve the shared goal (e.g., prepare and eat breakfast) with each of our two options: REACT and the baseline. To evaluate **H2**, we designed **Exp2** in which we randomly selected 10 configurations (from the 100 in **Exp1** and saved the corresponding answer sets (with REACT, baseline) to provide ground truth. Then, we posed 32 different questions (divided between the four types of questions) about some chosen steps in each trial corresponding to one of these 10 configurations, with answers computed as described in Section 3.3. We recorded the precision and recall of retrieving literals to answer these questions, with saved answer sets used to provide ground truth. Furthermore, we considered execution traces as qualitative evaluation of **H2**.

4.2 Experiment Results

Table 2 summarizes the results of **Exp1**. Since the plan length and task completion time will vary substantially between trials based on the configuration, we computed the values of the performance measures for REACT as a fraction (i.e., ratio) of those for the baseline in each trial. The average of these ratios is reported in Table 2. We observe that REACT reduces the number of steps and the time taken to achieve the goal compared with the baseline. This significant improvement in performance provided by REACT supports **H1**.

Table 3 shows the prediction accuracy of the behavior models learned by the ad hoc agent for the human using the ensemble of FF trees, as described in Section 3.2. We observe that the model does make errors, but it supports rapid learning and revision. Also, reasoning with prior knowledge and the output of

Table 3. Prediction accuracy of the learned behavior model for human.

Model	**Accuracy**
Human	85.19%

Table 4. Precision and recall of retrieving relevant literals for providing explanations.

Question type	*Precision*	*Recall*
Action justification	1.00	1.00
Contrastive	0.89	0.99
Belief justification	0.88	0.94
Counterfactual	1.00	0.78

these predictive models significantly improves the performance (as summarized above). In other experiments not reported here due to lack of space, we noticed that the ad hoc agent was able to revise these models to both improve accuracy and to track (or adapt to) changes in the human's behaviors over time.

Table 4 summarizes results of **Exp2**. The observed high values of precision and recall indicate the ability to automatically extract the correct literals to provide relational descriptions as explanations in response to different types of questions, thus supporting hypothesis **H2**.

4.3 Execution Trace

Next, we provide execution traces illustrating REACT's capabilities. Consider the scenario in which the: bread slice is inside toaster; cutlets are on kitchen counter; poundcake is on kitchen table; water glass is in bedroom; microwave is closed and switched off; frying pan is on stove that is switched off; and the human and ad hoc agent are in the kitchen. To help the human prepare breakfast, the ad hoc agent generated a plan with 23 steps, some of which are shown in Table 5; the agent expects the human to complete some intermediate steps.

Execution Example 1. [Action Justification, Contrastive, Counterfactual] Consider an exchange with the ad hoc agent after executing first plan step.

- **Questioner:** "Why did you find bread slice in step 0 ?"
- **Ad hoc Agent:** "Because I had not found the bread slice yet and I wanted to grab it in step 1".
  The agent's response draws attention to the target action's outcome being a requirement for executing a subsequent action. The agent can also be asked why it did not consider picking up a different object.

Table 5. Part of the ad hoc agent's plan for task in Execution Example 1; *brdslice* refers to breadslice, *mcrwave* to microwave, *ktchntable* to kitchentable.

occurs(find(ahagent,brdslice),0)	occurs(putin(ahagent,cutlets,fryingpan),8)
occurs(grab(ahagent,brdslice),1)	occurs(put(ahagent,cutlets,ktchntable),11)
occurs(put(ahagent,brdslice,ktchntable),2)	occurs(grab(ahagent,poundcake),13)
occurs(switchon(ahagent,stove),4)	occurs(open(ahagent,mcrwave),15)
occurs(grab(ahagent,cutlets),6)	occurs(put(ahagent,poundcake,mcrwave),16)

Table 6. Part of plan generated by ad hoc agent to avoid conflict with human.

occurs(find(ahagent,cutlets),1)	occurs(put(ahagent,poundcake,microwave),9)
occurs(grab(ahagent,cutlets),2)	occurs(close(ahagent,microwave),10)
occurs(find(ahagent,fryingpan),3)	occurs(switchon(ahagent,microwave),11)
occurs(putin(ahagent,cutlets,fryingpan),4)	occurs(open(ahagent,microwave),13)
occurs(grab(ahagent,poundcake),6)	occurs(put(ahagent,poundcake,kitchentable),16)
occurs(open(ahagent,microwave),8)	occurs(put(ahagent,cutlets,kitchentable),24)

- **Questioner:** “Why did you not find the water glass in step 0?"
- **Ad hoc Agent:** “Because I predicted that the human will find the water glass in step 0."

  The agent may also be asked about the human’s action choices.
- **Questioner:** “Why do you think human will grab water glass in step 1?"
- **Ad hoc Agent:** “Because my prediction is that the human wants to bring the glass to the table."
- **Questioner:** “What if the human decided to grab the cutlets in step 1 ?"
- **Ad hoc Agent:** “If the human grabs the cutlets in step 1, they will be in the human’s hands in step 2."

  To answer questions about hypothetical situations, the ad hoc agent has to simulate the evolution of state, and the execution of actions by the human and the agent, for a few steps in order to identify and report the motivation for specific action choices.

Execution Example 2. [Action Justification, Contrastive, Belief Tracing] The accuracy of the model predicting human action choices (85%, Table 3) indicates that the predictions can be incorrect, particularly in situations in which the agent’s understanding of the human is limited to a few observations. Consider a variant of the scenario above, in which the human decided to find and grab a slice of bread as the first action. Since the ad hoc agent decided to do the same action, it created a conflict. Table 6 shows key steps of the plan computed by the ad hoc agent to overcome this conflict. The subsequent conversation between the questioner and the ad hoc agent is as follows:

- **Questioner:** “Why did you find cutlets in step 1 ?"
- **Ad hoc Agent:** “Because I have not found the cutlets and I wanted to grab them in time step 2".
- **Questioner:** “Why did you not grab bread slice in step 1?"
- **Ad hoc Agent:** “Because the human was holding the bread slice."

  This exchange demonstrates the ability of the ad hoc agent to change its plan in order to prevent a conflict with the human and to justify this decision. Figure 4 shows the belief tree created for tracing the ad hoc agent’s beliefs and justifying why it did not grab the bread slice in step 1. The green and blue boxes represent the satisfied axioms and their grounded literals while red boxes represent the axioms that were not satisfied in this example.

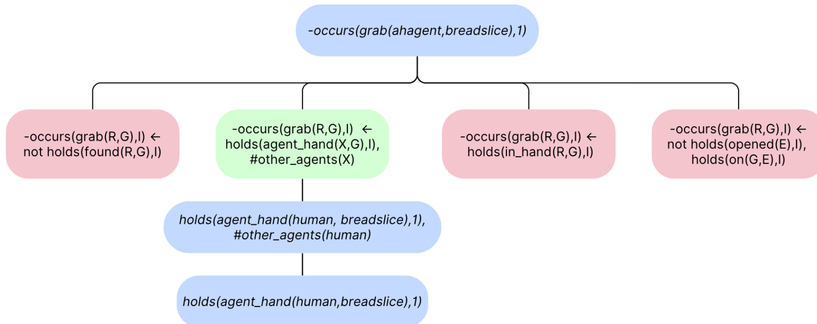

Fig. 4. Belief tracing to explore why ad hoc agent did not grab bread slide in step 1.

Overall these results support hypothesis **H2**. Additional results in the form of videos are available in our open-source code repository [10].

5 Conclusions

This paper described REACT, an AHT architecture for an embodied AI agent to collaborate with a human by reasoning with prior commonsense domain knowledge and incrementally learned models predicting the behavior of the human. REACT combines the principles of non-monotonic logical reasoning and ecological rationality, automatically identifying and reasoning efficiently with the relevant information. Also, the interplay between reasoning and learning enables the embodied AI agent to provide relational descriptions as on-demand explanations of its own decisions and those of the human. Experimental evaluation in a realistic physics-based simulation environment (VirtualHome) demonstrates REACT's capabilities and the improvement in performance compared with a logical reasoning baseline. Future work will explore: (a) incremental learning of other aspects of domain knowledge (e.g., axioms) in such complex domains; (b) scalability to multiple ad hoc agents and humans; and (c) implementation and evaluation on physical robots in AHT settings.

Acknowledgments

This work was supported in part by the U.S. Office of Naval Research Award N00014-20-1-2390. All conclusions are those of the authors alone.

References

1. Anjomshoae, S., Najjar, A., Calvaresi, D., Framling, K.: Explainable agents and robots: Results from a systematic literature review. In: International Conference on Autonomous Agents and Multiagent Systems. Montreal, Canada (2019)
2. Balai, E., Gelfond, M., Zhang, Y.: Towards Answer Set Programming with Sorts. In: Conference on Logic Programming and Nonmonotonic Reasoning (2013)
3. Balduccini, M., Gelfond, M.: Logic Programs with Consistency-Restoring Rules. In: AAAI Spring Symposium on Logical Formalization of Commonsense Reasoning (2003)

4. Baral, C., Gelfond, G., Pontelli, E., Son, T.C.: An action language for multi-agent domains. Artificial Intelligence **302**, 103601 (2022)
5. Barrett, S., Stone, P., Kraus, S., Rosenfeld, A.: Teamwork with limited knowledge of teammates. In: AAAI Conference on Artificial Intelligence. vol. 27 (2013)
6. Bowling, M., McCracken, P.: Coordination and adaptation in impromptu teams. In: National Conference on Artificial Intelligence. p. 53–58 (2005)
7. Chen, S., Andrejczuk, E., Cao, Z., Zhang, J.: AATEAM: Achieving the ad hoc teamwork by employing the attention mechanism. In: AAAI (2020)
8. Dodampegama, H., Sridharan, M.: Back to the Future: Toward a Hybrid Architecture for Ad Hoc Teamwork. In: AAAI Conference on Artificial Intelligence (2023)
9. Dodampegama, H., Sridharan, M.: Knowledge-based Reasoning and Learning under Partial Observability in Ad Hoc Teamwork. Theory and Practice of Logic Programming **23**(4), 696–714 (2023)
10. Dodampegama, H., Sridharan, M.: https://github.com/hharithaki/REACT (2024)
11. Fox, M., Long, D., Magazzeni, D.: Explainable Planning. In: IJCAI Workshop on Explainable AI (2017)
12. Gelfond, M., Inclezan, D.: Some Properties of System Descriptions of AL_d. Applied Non-Classical Logics, Special Issue on Equilibrium Logic and ASP **23**(1-2) (2013)
13. Gigerenzer, G.: What is Bounded Rationality? In: Routledge Handbook of Bounded Rationality. Routledge (2020)
14. Katsikopoulos, K., Simsek, O., Buckmann, M., Gigerenzer, G.: Classification in the Wild: The Science and Art of Transparent Decision Making. MIT Press (2021)
15. Mirsky, R., Carlucho, I., Rahman, A., Fosong, E., Macke, W., Sridharan, M., Stone, P., Albrecht, S.: A Survey of Ad Hoc Teamwork: Definitions, Methods, and Open Problems. In: European Conference on Multiagent Systems (2022)
16. Puig, X., Ra, K., Boben, M., Li, J., Wang, T., Fidler, S., Torralba, A.: Virtualhome: Simulating household activities via programs. In: International Conference on Computer Vision and Pattern Recognition. pp. 8494–8502 (2018)
17. Rahman, M.A., Hopner, N., Christianos, F., Albrecht, S.V.: Towards open ad hoc teamwork using graph-based policy learning. In: International Conference on Machine Learning. pp. 8776–8786 (18–24 Jul 2021)
18. Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., Parikh, D., Batra, D.: Habitat: A platform for embodied AI research. CoRR abs/1904.01201 (2019)
19. Son, T.C., Sakama, C.: Reasoning and Planning with Cooperative Actions or Multiagents using Answer Set Programming. In: Declarative Agent Languages and Technologies VII, LNCS, vol. 5948, pp. 208–227. Springer Berlin Heidelberg (2010)
20. Sridharan, M.: Integrated Knowledge-based Reasoning and Data-driven Learning for Explainable Agency in Robotics. In: Explainable Agency in Artificial Intelligence: Research and Practice. CRC Press (2024)
21. Sridharan, M., Mota, T.: Towards Combining Commonsense Reasoning and Knowledge Acquisition to Guide Deep Learning. Autonomous Agents and Multi-Agent Systems **37** (2023)
22. Stone, P., Kaminka, G., Kraus, S., Rosenschein, J.: Ad Hoc Autonomous Agent Teams: Collaboration without Pre-Coordination. In: AAAI Conference on Artificial Intelligence. pp. 1504–1509 (2010)
23. Zand, J., Parker-Holder, J., Roberts, S.J.: On-the-fly strategy adaptation for adhoc agent coordination. In: International Conference on Autonomous Agents and Multiagent Systems. p. 1771–1773 (2022)