TeamCollab: A Framework for Collaborative Perception-Cognition-Communication-Action

Julian de Gortari Briseno *UCLA*
Roko Parać* *ICL*
Leo Ardon* *ICL*
Marc Roig Vilamala *Cardiff University*
Daniel Furelos-Blanco *ICL*
Lance Kaplan *DEVCOM ARL*

Vinod K. Mishra *DEVCOM ARL*
Federico Cerutti *University of Brescia*
Alun Preece *Cardiff University*
Alessandra Russo *ICL*
Mani Srivastava *UCLA*

***Abstract*—Teams of embodied AI-enabled agents are critical for applications in extreme and highly dynamic environments. Developing robust controllers for such agents requires a deep understanding of the challenges encountered when attempting to coordinate and synchronize their individual perception-cognition-communication-action (PCCA) loops for team-wide mission objectives. We introduce a framework to explore the coordination of the PCCA loops across multiple agents in a new simulated physical environment designed to explore collaboration in each PCCA stage. This environment tasks teams of agents with the correct disposal of dangerous objects in an area and forces careful coordination of sensing, communication, movement, and manipulation actions by providing spatially-bounded communication, incorporating situations that require concerted effort by groups of agents, and introducing uncertainty into agents' sensing capabilities. We provide a set of heuristic controllers, an offline oracle model, and an initial exploration of a Reward Machine-based controller that learns its policies from training. Together these approaches serve to provide insights into the complexity of the multi-agent PCCA loop coordination problem. The multiagent PCCA simulation environment, which supports AI and human-controlled agents, and the code for various agent controllers are available at https://github.com/nesl/AI-Collab.**

I. Introduction

Driven by foundational advances in deep learning, availability of vast amounts of data to train on, and technological advances in hardware accelerators, embodied Artificial Intelligence (AI) agents have made considerable improvements in their sensing, perception, decision-making, and physical capabilities, often far exceeding humans. However, these improvements frequently do not translate into commensurate improvements when AI agents operate in teams with other AI agents and humans. Factors such as a lack of shared awareness of the situation, and a lack of understanding of team members' policies lead to poorly coordinated actions. Moreover, failure to accurately model partners in collaborative settings can lead to arbitrarily poor performance by the team [1]. The problem of miscoordination is exacerbated by the black box nature of large data-driven machine learning models that now power AI agents [2]. Effective collaboration between AI agents and humans requires careful consideration of the spatiotemporal properties of the real world.

Prior work has tended to evaluate cooperative AI performance under limited environments without dealing with the full synchronization of individual perception-cognition-communication-action (PCCA) loops with team-wide global PCCA loops that characterize collaboration in real-life scenarios. Within this pipeline, collaboration in perception only arises in partially observable environments where, for example, imprecise sensing may require multiple teammates to take measurements of the same entity to ascertain some property [3]. Equally important, collaboration in cognition has been approached in previous works as part of the Theory of Mind, i.e. modeling beliefs and internal mental states of other agents, either implicitly, such as by using traditional multi-agent reinforcement learning techniques that at training distill into each agent an understanding of their teammates' actions [1], [4], [5], or explicitly, by separately incorporating an external notion of the agents collaborative potential and the task at hand [6]–[8]. Communication has usually been included as an indirect exchange of information that is a product of teammates' behavior [9]. Direct exchange of information through messages has usually been targeted towards the resolution of low-level conflicts that arise when coordinating actuation [10], [11], but recent work has attempted communication between agents for high-level action planning [12]. Finally, while collaboration in actions is inherent to all cooperative AI work, different environments and tasks may enforce or relax spatiotemporal constraints on it, such as whether two or more agents need to perform the same action at the same time and whether they need to be at nearby locations to be able to succeed in it [13].

Our goal in this work is to provide a framework for the study of AI collaboration with teams of agents, taking into account the coordination of their full PCCA loops. Consequently, we provide a new environment based on an enhanced version of the ThreeDWorld physics simulator [14] that simulates a real-life scenario in which AI agents can interact simultaneously as part of teams of different sizes and compositions, allowing even humans to participate as teammates. We evaluate a set of 3 heuristic algorithms for controlling autonomous agents in this environment. These algorithms vary in the type of team structure and information fusion strategies they employ, such structures being taken from related work in the field of human teamwork [15]. Results quantify the limits of human

*Equal contribution

knowledge distillation with regards to an offline oracle method that provides an upper bound on the possible performance improvement. As a consequence, the promising first steps towards a learning method that incorporates a mix of human intuition are described in this work as well, to understand their potential impact. The contributions in this paper are a novel simulated environment for testing multi-agent collaboration in teams within the context of a full PCCA pipeline and a set of new approaches and evaluating mechanisms for such task (heuristic planning, an ideal oracle method, and the first steps towards learning methods with reward machines).

II. Related Work

Multi-agent collaboration in the context of PCCA loops has been partially studied in previous works. Collaboration in cognition has been an active field of research with works such as [1], where agents are trained with proxies of humans, and [4], [5] where agents are trained together with a diverse set of agents, themselves trained with different algorithms, play styles, and skill levels. These works and others in the field of multi-agent collaboration have relied on 2D environments without any physics simulation such as Overcooked [1], [4], [6], [16] and Hanabi [5], [17], that incorporate rules promoting collaborative behavior like in our environment, but operate in a more constrained setting that either entirely abstracts away the notion of embodiedness (Hanabi), or simply does not consider the uncertainty present in real-world perception and actuation (Overcooked) that our environment, built on top of the ThreeDWorld platform [14], does simulate.

In [8] the VirtualHome-Social environment simulates a 3D real-world indoor setting where an AI agent is tasked first to understand and then to help another agent in their endeavors. However, explicit communication is not included in their implementation, and the challenges of movement coordination for multiple agents that appear in reality are abstracted away. An improvement over this environment is shown in [12], with explicit cooperative communication mechanisms now included, but still without low-level motion planning and collaboration. Similarly, Habitat 3.0 [18] abstracts away these problems. Low-level motion planning, as required in our simulated environment, is important for real-world applications given the brittleness of robotic bodies and the importance of safeguarding human collaborators, thus accounting for these factors is necessary to more easily transfer novel controllers from the simulated environment to the real world. Multi-agent collaboration has also been explored at a lower level, with works that focus explicitly on motion coordination, perception coordination, or coordination in communication, relevant for swarms of UAVs like in [3], [19], or in the context of warehouse robots such as in [10], [11].

The heuristic controller approaches we present in this work have been influenced mostly by related work on human teamwork [15]. This was chosen mainly because of the complexity of the simulated environment, akin to the real world where human teams thrive, and the ultimate goal in future work of introducing human teammates. Related work has explored such human team frameworks in limited ways as in [20]. Finally, the learned approach we implement in this work is based on Reward Machines. Several lines of research have based their work on the concept of Reward Machine as a way to model non-Markovian rewards [21], [22]. In the multi-agent setting, the use of finite-state machines for task decomposition in collaborative teams of agents has often adopted a top-down approach to construct sub-machines from a global task (e.g. [23]). Unlike our work, some of these methods focus on the decomposition of the task but not on how to execute it. In this paper, we followed the multi-agent with RMs approach proposed in [13] whereby the structure of the task is known "a priori", agent-specific RMs are "pre-engineered" to capture the task decomposition among multiple agents, and the RMs of the different agents are executed in parallel.

III. A Framework for PCCA Collaboration

Effective collaboration between agents' PCCA loops requires considering the collaborative potential in each dimension separately as shown in Figure 1. In our simulated environment (Figure 2), a team of robotic agents is tasked with the mission of moving all dangerous objects to a goal area in the least amount of time, relying on: **(1)** sensors to detect whether an object is dangerous or not, **(2)** communication between agents to exchange information, **(3)** whatever help agents might provide to each other to carry objects, and in general, **(4)** the ability of agents to fuse information, assign themselves tasks and coordinate movement. In the following sections, an explanation will be given for each of the stages of the PCCA loop, as well as details of the simulation platform.

A. Perception

Perception in our simulated environment is achieved through two sensors: a camera that provides a first-person view of the scene from the perspective of the robotic avatar being controlled, and an abstract danger sensor that allows any agent to measure whether any object within a given radius from the agent is dangerous or not. Agents are also able to choose to replace the first-person camera view with an occupancy map of the scene that is also limited in its view.

All objects in the scene have an associated true danger status that is decided at scene initialization. This is a binary value that determines whether an object is benign or dangerous. Danger sensors are defined as binary asymmetric channels with two parameters that establish the probability of truly and falsely detecting a benign object and the probability of truly and falsely detecting a dangerous object. These two parameters vary per agent, are obtained initially from a uniform distribution, and together describe the confidence of an agent over their sensor measurements. A weighted random choice based on the previous parameters is made whenever a new object is sensed, repeated sensing of the same object resulting in the same output. Given that sensor measurements are uncertain, no single agent can distinguish all the truly dangerous objects that agents are required to dispose of, forcing collaboration.

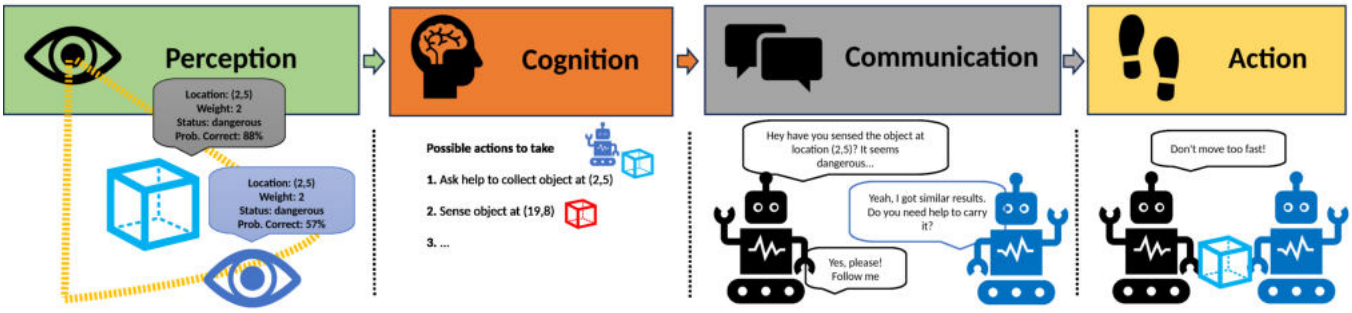


Fig. 1: Collaboration in all stages of the team-wide Perception-Cognition-Communication-Action (PCCA) loop is exemplified in this figure for our simulated environment. Agents exchange estimates of their sensing results to find those that are truly dangerous (**Perception**), reason about collaborative actions to take (**Cognition**), communicate to share information and propose collaboration (**Communication**), and finally, coordinate their movement to carry objects to the goal area (**Action**).

B. Cognition

Each agent is controlled separately by a different algorithm that may be plugged in dynamically before the scene starts. Humans may also participate in the control of one of the robotic avatars. In this particular work, a set of heuristic algorithms (Section V) and a learning method (Section VII) are utilized to explore collaboration at the cognitive level.

C. Communication

Agents can communicate with each other through a text-based chat mechanism, and send messages to all other agents within a given spatial radius. The specific value of this parameter may constrain communication between agents to be spatially bounded or unlimited in its reach. Communication is intended to help agents coordinate actions, such as movement into adjacent spaces, as well as to allow agents to ask for help to carry heavy objects or to confer information about unreliable sensor measurements. Communication can also be used to share general status updates and issue orders.

AI agents communicate using a Controlled Natural Language (CNL) that is designed for the task. They use a predefined set of machine-processable templates that can be read by humans as natural language sentences and be filled out with relevant information as appropriate. For example, to ask for help, agents use the next template: "*I need **[num]** more robots to help carry object **[ID]***", where **[num]** is the number of requested agents, and **[ID]** the object ID. Using a CNL allows AI agents to communicate with each other unambiguously while making it easy for humans to interpret their messages, thus facilitating the interactions between humans and AI agents for future work [24]. However, future work will explore designing agents with free-form Natural Language capabilities.

D. Action

Agents can freely move around the simulated scene, get close to objects of interest, activate their sensors to obtain measurements, decide to carry and drop objects, and exchange messages with other agents. Besides each object having a danger status associated with it, a weight is also assigned randomly at scene initialization. This weight indicates the minimum number of agents required to carry the object. Collaboration for carrying objects is achieved in practice by making a single agent carry the object while the other agents stay within a certain radius of it (strength contribution radius). In effect, each agent has a strength variable that starts at one and temporally increases by one for each nearby agent. All agents need to make sure their current strength matches or surpasses the weight of the object to be lifted. If already carrying an object, the strength value needs to be maintained otherwise the object is automatically dropped.

E. Simulation Platform

The simulated environment utilizes an enhanced version of the ThreeDWorld physics simulator [14] we developed. It allows for multiple robot controllers or human operators to connect independently to the simulator through a network and interact in the same scenario. As shown in Figure 2, the scenario consists of an area surrounded by walls and four rooms. A goal area is delimited by a predetermined radius from the center of the scenario, the task at hand is to bring all truly dangerous objects to the goal area in the least amount of time. The simulated environment runs in a server implemented using Node.js, with communication between the server and remote clients achieved using Socket.IO and WebRTC.

IV. Problem Definition

The problem being faced by agents in the simulated environment can be defined as a decentralized partially observable Markov decision process (Dec-POMDP) augmented with communication, formalized by the tuple $(S, \{A_i\}, T, R, \{\Omega_i\}, O, \gamma)$, where S is the set of states, A_i is the set of actions for agent i, T is the set of conditional transition probabilities between states, R is the reward function, Ω_i is the set of observations for agent i, O is the set of conditional observation probabilities, and γ the discount factor. The set of states corresponds to the location and properties of each agent and object in the scene, as well as messages exchanged. Individual agents only have partial observability of

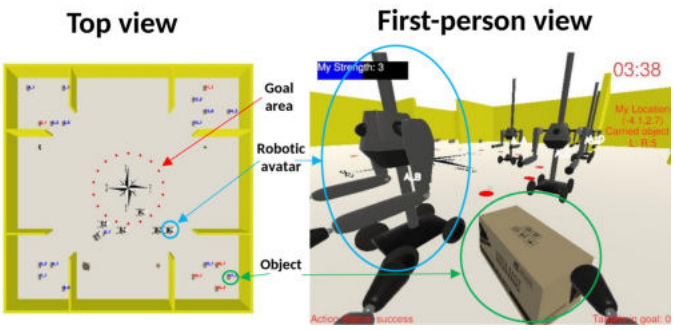


Fig. 2: A top view and first-person view of the simulated environment. Agents have access to the first-person view of the avatar they control, and according to the experimental conditions enforced, may be given access to the top view.

this state, although the locations of objects and other agents are fully observable to each agent. The properties associated with each agent correspond to their strength value, the uncertainty of their sensors, and the results of their sensing actions. The properties associated with each object are their weight and danger status. Actions are limited to the ones described in Section III-D. To solve this problem, a set of symbolic heuristic planners are proposed as described next.

V. Heuristic Strategies

Our heuristic strategies are composed of a series of symbolic methods distilled completely from human knowledge and defined through state machines that try to explore the space of possible collaborative strategies designed for efficient solving of the task at hand. These heuristic strategies are based on high-level, human-only, task-independent forms of organization documented and classified in previous works that we are now utilizing to evaluate their performance within the scope of our task [15]. Future work involving human-AI teams may benefit from this choice given that AI agents are adopting a collaborative framework understandable to humans. While human teamwork can assume many forms, we were compelled to study two dimensions of team structure that have the most impact on our current task and have been explored in simpler environments for AI-only teams [20]: role structure and leadership structure. Dimensions such as communication structure are left for future work. Heuristic strategies based on variations in these two dimensions are distinguished mainly by the degree of autonomy given to agents and the distinct capabilities allocated to each agent, our objective being to study how well each of these strategies minimizes the task's completion time and achieves a high score.

Each team structure dimension provides us with a set of discrete classes to explore: for leadership structure, collaboration between agents may follow a hierarchical structure, with a single agent being the designated leader from start to finish, working as a centralized planner that collects information provided by other agents and allocates tasks to them. Alternatively, in a more decentralized approach, agents will try to autonomously come up with their next tasks, share information with any agent they find nearby, and temporarily become leaders to direct teams of agents when needed. With regards to the other dimension of team structure explored, role structure, collaboration may follow a functional form of labor approach where agents have specialized roles throughout the task, as opposed to a more divisional approach, where agents are capable of performing all kinds of roles at any time. For the current task in our simulated environment, agents may choose to play one of two distinct roles: a scout, sensing objects in the scene and sharing their measurements with other agents, or a lifter, where their only task is to carry those objects they know may be dangerous to the goal area. Notwithstanding the type of heuristic strategy used, they all utilize the same low-level building blocks that allow for movement coordination.

Three concrete heuristic strategies are explored in this work: designated leadership with divisional roles (**Designated**), where a single agent is assigned the role of leader and assigns tasks to other agents who can perform any type of role, temporary leadership with functional roles (**TempFunc**), where the leadership function is rotated across time and tasks between agents that have fixed roles, and temporary leadership with divisional roles (**TempDiv**), where leadership is similarly rotated but with agents that can perform any type of role.

At a high level, for every heuristic approach, agents seek to gather information about objects' danger status before trying to carry them. In all strategies, agents exchange information about their sensing results with other agents. Based on their confidence in such results, fusion is achieved by taking those measurements with the highest confidence level. Other fusion strategies are available in our environment but not explored within this work. Agents keep a register of all information being communicated by nearby agents so that whenever any object's information gets updated, they will communicate such information to those agents they encounter that have not been registered receiving it. To enable agents to traverse a dynamic scenario, a shortest path is computed at each time step as needed. Coordination with other agents for movement is based on previous works [10], [11], achieved in practice by making agents communicate with those within range and solving any path conflict that may arise by following a fixed set of right-of-way rules. Finally, coordination for carrying an object with a weight greater or equal to two is achieved by having an agent announce their desire to carry such a heavy object and request help. If other agents respond to such a request, a squad gets formed, where the agent who requested help becomes the squad leader and has the authority to make other squad members move according to its needs for the duration of time it takes to dispose of the desired object.

A. Evaluation

Our heuristic algorithms are evaluated inside our environment for their capability to optimize performance of a team of agents seeking to collect only all truly dangerous objects in the scene. We devised a formula to evaluate team's performance,

consisting of quality and speed factors. For each agent i, their Individual Quality of Work (IQW) is defined as:

$$IQW[i] = \max\left(0, \frac{O_{d_{Col}}[i] - O_{nd_{Col}}[i] - O_{d_{Drop}}[i]}{N_{d_{obj}}}\right) \quad (1)$$

where $O_{d_{Col}}$ corresponds to the truly dangerous objects collected, $O_{nd_{Col}}$ to the truly non-dangerous objects collected, $O_{d_{Drop}}$ to the truly dangerous objects dropped accidentally, and $N_{d_{obj}}$ to the total number of truly dangerous objects in the scene. This formula penalizes agents for not carefully choosing the type of objects being collected and for not coordinating correctly with their teammates. The Team Quality of Work (TQW) is a summation of the IQW of all agents. This formula gives us an understanding of the aggregate quality of actions taken by all agents. Concerning the speed factor, the Team Speed of Work (TSW) is determined by:

$$TSW = \frac{T_{timeout}}{\max(T_{timeout}/10, \min(T, T_{timeout}))}, \quad (2)$$

where $T_{timeout}$ is a predetermined timeout value, after which the session ends automatically ($T_{timeout} = 20min$ in our experiments), and T is the actual ending time. While agents are not penalized for using the whole time allocated for the overall task, any improvement in time completion over the time limit gets a corresponding bonus. Additionally, a limit is imposed on how fast a session may end and its corresponding award. Based on the quality of work and speed of work, the final team score is computed as:

$$TeamScore = TQW \cdot TSW. \quad (3)$$

The team score allows us to evaluate our algorithms and compare them with each other, as will be shown in the next subsection. Other metrics utilized to evaluate our algorithms are the fraction of dangerous objects collected over the total number of dangerous objects existing in the scene, as well as the fraction of dangerous objects collected concerning the total number of objects collected, both of which let us understand better the factors that influence the team score. Finally, a metric defined by the average number of times sensors are activated allows us to measure the effectiveness of coordination between agents based on their need to individually collect redundant measurements. We leave for future work other metrics related, for example, to the amount and content of messages exchanged between team members.

B. Experiments

Table I shows the values of some of the simulation parameters we used for our evaluation. Each of our heuristic algorithms was evaluated with a team size of 5 agents for 40 sessions of 20 minutes each, enough time to complete the task. Such a team size was chosen to compare our results with future work that will include teams of humans. Since the performance of the simulation platform is degraded with an increasing number of simulated agents, 5 is a good compromise. In each session, 20 objects are instantiated inside a scenario of 20×20 meters, 5 objects being assigned per room. The starting positions of both agents and objects within a room are randomly chosen, as well as the parameters that characterize the confidence over each agent's sensors, and the true danger status and weight of each object. When objects get instantiated, a dangerous status is only assigned to an object with a probability of 0.3, and weights are assigned to objects with probabilities that follow the next recursive formula: $w_n = w_{n-1}/2$. If $w_0 = 1$, a weight of 1 (w_1) will be assigned with 0.5 probability, a weight of 2 with 0.25, and so on. The maximum weight an object can be assigned equals the number of agents in the scene. Results are shown in Figure 3 for the different heuristic strategies. For the **TempFunc** strategy, two agents are assigned the role of scouts and three agents the role of lifters, although the first agents may become lifters after having sensed all objects.

TABLE I: Simulation Parameters.

Parameter	Value
Team Size	5
Number of Objects	20
Session Duration	20 min
Occupancy Map Cell Size	1 m^2
Sensing Radius	2 m
Strength Contribution Radius	3 m
Communication Radius	5 m
Goal Radius	3 m

C. Discussion

Our results show that heuristic strategies converge around an average team score value of 0.3, notwithstanding the different team structures. While these team structures are designed to reduce completion time and achieve a high team score, they seem to fail to effectively use agents' collective potential. The more hierarchical approach, the **Designated** team structure, can reduce the number of total sensor activations per session without necessarily impacting the overall team score, showing the benefits of having a single point of information fusion, contrary to the **TempDiv** structure, where agents roam freely around the scene wasting time in redundant sensing. Similarly, the more unstructured nature of **TempDiv** allows its agents to maximize the collection of objects to the detriment of their true danger status, contrary to both **Designated** and **TempFunc** that place a greater emphasis on sensing and fusion of results but lose on the total amount of collected dangerous objects. The similarity in team scores, despite different team structures, suggests that shared low-level strategies derived from heuristics may be the issue. Moreover, given the task's dynamic nature, one cannot assume a single heuristic will be valid at all points in time. Finally, a Team Speed of Work of 1 was obtained in all cases, demonstrating that teams could not finish before the time limit.

VI. Idealized Oracle Strategy

We also explored an idealized strategy where the agents are oracles who a priori know the objects to be collected, to understand the upper-performance limits, and how close our

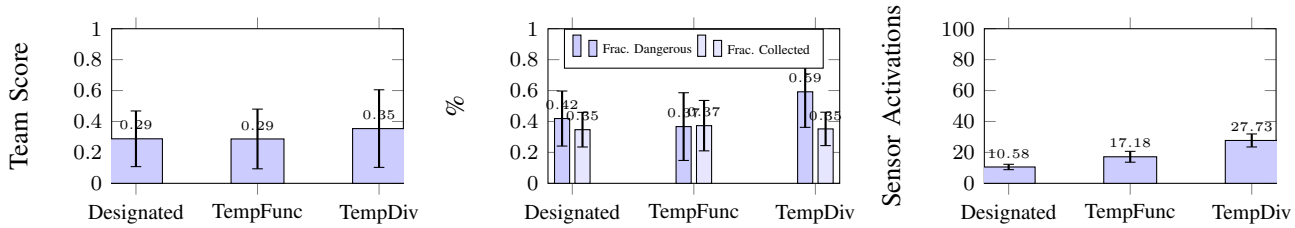


Fig. 3: For each of the heuristic algorithms evaluated, the following metrics are reported: average team score; average fraction of dangerous objects collected to the total number of dangerous objects (Frac. Dangerous), and the total number of collected objects (Frac. Collected); and, the average number of sensor activations. Error bars show the standard deviations.

heuristic strategies are to those limits. A team of such agents does not need to spend any time sensing or communicating, and can instead compute upfront an optimal plan for collecting and disposing of the dangerous objects. The task at hand then is simply to statically decide which agents will collect and transport which object at what time.

This problem is similar to the optimization problem known as Parallel Task Scheduling that can be adapted to our setting in the following way: there is a set O of n objects and m agents with identical capabilities. Each object $o \in O$ has an associated time t_o needed for correct disposal and requires the simultaneous effort of q_o agents. A schedule must assign for the collection of each object o a starting time s_o and a set $m_o \subseteq \{1, ..., m\}$ of $|m_o| = q_o$ agents that will help in such task. The objective is to find the schedule with the minimum length or makespan $C_{\max} = \max_{o \in O}(s_o + t_o)$. The time associated with the correct disposal of each object is calculated as a measure of the number of time steps needed to move from the goal area to the object in question, pick it up, and drop it in the goal area. To calculate the optimal path needed to make the traversal to each object and the associated time steps, the A* shortest path algorithm is utilized. The exact implementation of the scheduling algorithm is based on [25].

A. Evaluation

Our offline oracle strategy was utilized to compute the optimum schedule given 40 different arrangements of objects and danger status assignments, compliant with the environment characteristics described in Section V-B. In Table II, a comparison is made between the oracle strategy and the heuristic strategies concerning three metrics: Team Quality of Work, distance traveled, and productivity. Given full observability of the environment, agents' movement should be optimized such that the total distance traveled by all agents will be less than any other strategy that relies on first discovering the danger status of each object and has to deal with the uncertainty of moving unnecessary objects. Similarly, productivity is defined as the number of truly dangerous objects collected divided by the distance traveled, measuring the quality of agents' planning capabilities with respect to the overall goal.

B. Discussion

The Team Quality of Work differences quantify the ample room for improvement that still exists between the heuristic strategies and an ideal strategy. The productivity results show again that **TempDiv** is the strategy that minimizes the most the idle time of individual agents, by letting them act in a fully autonomous manner to carry any object they find, as opposed to the other methods where agents wait for sensing results from their teammates before trying to carry any object. More importantly, the offline oracle establishes a lower bound on distance traveled that is half what the heuristic planners achieved, and a higher bound on productivity, with a result that is around four times what any of our algorithms achieved.

VII. Learning Policies with Reward Machines

The heuristic strategies presented in Section V contain a distinct human bias in the form of hard-coded logic that limits teams of agents' generalization capabilities and scalability to dynamic environments. However, such inadequacy can be abated by incorporating a learning mechanism into agents' planning strategies. Particularly, through reinforcement learning (RL), agents can learn from their experiences and ideally adapt to any environment. By also including human knowledge, the learning process can be accelerated.

While useful to mitigate the bias introduced by hand-coded heuristics, RL-based approaches lack interpretability and correctness guarantees, two characteristics often crucial in high-stakes real-world applications. To this end, we propose utilizing a reinforcement learning approach based on Reward Machines (RMs) — finite-state automata that represent the reward function of a task in terms of high-level events. In the single-agent setting, the RM approach has shown to be effective for modeling non-Markovian rewards, for learning policies that generalize across different instances of a task [26], and for supporting re-usability of learned policies across tasks of increased complexity [27]. In the context of partial observable settings, the application of RMs to multi-agent cooperative teams has also shown promising results in grid world domains [13], [28], where the global (team-level) reward function is decomposed into agent-specific RMs that capture the respective agent's sub-task and relevant teammate communication for successful cooperation.

In the context of this paper, we conducted an initial exploration of the effectiveness of RMs for the multi-agent PCCA coordination problem. The realism provided by the simulator poses a few challenges for agents' learning given that they

TABLE II: Comparing Heuristic Strategies with Offline Oracle

Strategy	Team Quality of Work	Difference	Distance Traveled (m)	Gain	Productivity (objects/m)	Gain
Oracle	1.0	0	360.95	x1.0	0.0193	x1.0
Designated	0.288	0.712	553.24	x1.53	0.00428	x0.22
TempFunc	0.287	0.713	552.80	x1.53	0.00425	x0.22
TempDiv	0.354	0.646	577.77	x1.60	0.00640	x0.33

now have to deal with real-time and asynchronous events. In this section, we first consider a simplified version of the simulated environment that does not incorporate the physics engine. We allow for full visibility of objects' locations and true danger status, and for the environment to contain only two agents and one object of weight equal to two. We trained each agent separately in the simplified environment using the RM algorithm in [28], and then evaluated the performance of the learned agents' policies in the physics-enabled simulated environment, as shown in Figure 4, demonstrating how the joint execution of the agents' policies leads to a team-level solution of the global task.

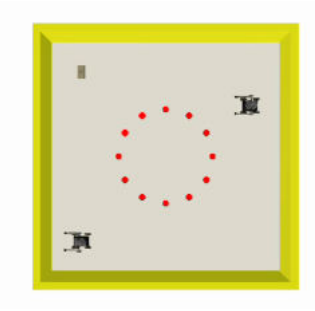
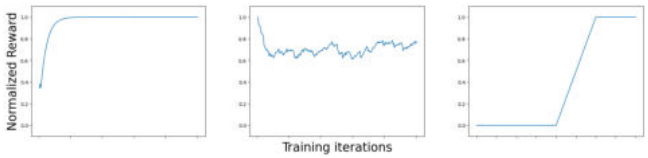


Fig. 4: The physics-enabled simulated environment contains only two collaborating agents and one object. The goal is to carry that object into the goal location.

A. Experiments

The RL agents need to find the policy $\pi(a|s)$ that maximizes the γ-discounted total return ($R = \sum_{k=0}^{T} \gamma^k r_k$) of an episode, where r_k represents the per-timestep reward obtained by the team of agents, and T the maximum allocated time to solve the task. Agents can perform the following actions: *do_nothing*, *up*, *left*, *right*, *down*, *pick_up*, *drop*. The state s contains environment information, such as the position of each agent and object in the scene, as well as sensing results of objects close to an agent. The immediate reward at a time t is $r_t = 10$ if a dangerous object is dropped in the goal area at that step. Otherwise, $r_t = -0.01$. Individual agents are given predefined RMs that model the sub-tasks needed to complete the overall high-level task. Each agent is trained using the Q-learning for Reward Machines (QRM) algorithm [21]. Events requiring coordination of multiple agents are generated randomly with a given probability during the isolated training phase of each individual agent so that they can still learn how to accomplish their task without being impacted by the quality of policies from the other agents.

Figure 5a and 5b show the reward obtained by each agent during their isolated training inside the simplified environment. Figure 5c shows the collective reward obtained at test time in such an environment, from which we can infer that the team of agents successfully learns to collaborate. We then evaluated the performance of the policies trained in the simplified environment with the physics-enabled simulated environment over multiple runs and observed a 60% success rate for agents moving the dangerous object into the goal location within a time limit. Failures arose mainly from the stochastic nature of the physics engine affecting agents when carrying objects.



(a) Agent 1 Reward (b) Agent 2 Reward (c) Team Reward

Fig. 5: Training performance of RMs within the simplified environment

B. Discussion

Preliminary results from the experiments demonstrate the potential benefits of using RL with RMs for the coordination of PCCA loops within multi-agent teams. We hypothesize that the abstract level of representation and decomposition into sub-tasks that RMs offer can help mitigate the problem of transferring the policies trained in the simplified environment to the physics-enabled simulated environment while keeping acceptable levels of performance.

Despite the promising results, the experiments were conducted in a simplified version of the environment. Moreover, several challenges must be addressed. First, the assumption on RMs being handcrafted: the approach needs to be extended to interleave the learning of the RMs associated with each agent with the learning of their associated RL policies, whilst guaranteeing that their executions achieve the global cooperative task (e.g. [28]). Secondly, the mapping from observations to high-level events is assumed to be perfect, whereas in real-world settings sensor readings are error-prone; thus, policy and RM learning algorithms that are robust to noise must be devised. Thirdly, we can potentially drive the RMs' abstraction power further by considering first-order logic formulae. Finally, function approximation methods such as DDPG and DQNs [22] should be embraced to enable generalization.

VIII. Conclusion and Future Directions

In this work, we presented a simulated environment and a set of initial strategies for solving the problem of multi-agent PCCA loop coordination. However, we intend this work to serve only as an initial step towards solving such a complex problem. In the future, methods that incorporate better knowledge about the underlying physics of the environment, that include a richer scheme of communication between agents and accounts for better explainability will be necessary to improve over our baselines. A more sophisticated handling of uncertainty in sensing measurements will be needed to coordinate sensing actions more effectively. Finally, our simulated environment already allows for human control of robotic avatars, so future work should explore the issues that arise with mixed human-AI teams, such as asymmetry in perception and cognition, obstacles of natural language communication, problems of trust calibration, and varying team structures.

Acknowledgment

The research reported in this paper was sponsored in part by the DEVCOM Army Research Laboratory via cooperative agreement W911NF2220243. 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 United States government.

References

[1] M. Carroll, R. Shah, M. K. Ho, T. Griffiths, S. Seshia, P. Abbeel, and A. Dragan, "On the Utility of Learning about Humans for Human-AI Coordination," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 5175–5186.

[2] G. Marcus, "The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence," *arXiv preprint*, vol. arXiv:2002.06177, 2020.

[3] J. Hu, H. Zhang, L. Song, R. Schober, and H. V. Poor, "Cooperative Internet of UAVs: Distributed Trajectory Design by Multi-Agent Deep Reinforcement Learning," *IEEE Transactions on Communications*, vol. 68, no. 11, pp. 6807–6821, 2020.

[4] D. Strouse, K. McKee, M. Botvinick, E. Hughes, and R. Everett, "Collaborating with Humans without Human Data," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2021, pp. 14 502–14 515.

[5] K. Lucas and R. E. Allen, "Any-Play: An Intrinsic Augmentation for Zero-Shot Coordination," in *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2022, pp. 853–861.

[6] S. A. Wu, R. E. Wang, J. A. Evans, J. B. Tenenbaum, D. C. Parkes, and M. Kleiman-Weiner, "Too Many Cooks: Bayesian Inference for Coordinating Multi-Agent Collaboration," *Topics in Cognitive Science*, vol. 13, no. 2, pp. 414–432, 2021.

[7] D. Nguyen, S. Venkatesh, P. Nguyen, and T. Tran, "Theory of Mind with Guilt Aversion Facilitates Cooperative Reinforcement Learning," in *Proceedings of the Asian Conference on Machine Learning (ACML)*, 2020, pp. 33–48.

[8] X. Puig, T. Shu, S. Li, Z. Wang, Y. Liao, J. B. Tenenbaum, S. Fidler, and A. Torralba, "Watch-And-Help: A Challenge for Social Perception and Human-AI Collaboration," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[9] C. Liang, J. Proft, E. Andersen, and R. A. Knepper, "Implicit Communication of Actionable Information in Human-AI Teams," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 2019, pp. 1–13.

[10] A. Maoudj and A. L. Christensen, "Decentralized Multi-Agent Path Finding in Warehouse Environments for Fleets of Mobile Robots with Limited Communication Range," in *Proceedings of the International Conference on Swarm Intelligence (ANTS)*, 2022, pp. 104–116.

[11] H. Wang and M. Rubenstein, "Walk, Stop, Count, and Swap: Decentralized Multi-Agent Path Finding With Theoretical Guarantees," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1119–1126, 2020.

[12] H. Zhang, W. Du, J. Shan, Q. Zhou, Y. Du, J. B. Tenenbaum, T. Shu, and C. Gan, "Building Cooperative Embodied Agents Modularly with Large Language Models," *arXiv preprint*, vol. arXiv:2307.02485, 2023.

[13] C. Neary, Z. Xu, B. Wu, and U. Topcu, "Reward Machines for Cooperative Multi-Agent Reinforcement Learning," in *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2021, pp. 934–942.

[14] C. Gan, J. Schwartz, S. Alter, D. Mrowca, M. Schrimpf, J. Traer, J. D. Freitas, J. Kubilius, A. Bhandwaldar, N. Haber, M. Sano, K. Kim, E. Wang, M. Lingelbach, A. Curtis, K. T. Feigelis, D. Bear, D. Gutfreund, D. D. Cox, A. Torralba, J. J. DiCarlo, J. Tenenbaum, J. H. McDermott, and D. Yamins, "ThreeDWorld: A Platform for Interactive Multi-Modal Physical Simulation," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, 2021.

[15] J. L. Wildman, A. L. Thayer, M. A. Rosen, E. Salas, J. E. Mathieu, and S. R. Rayne, "Task Types and Team-Level Attributes: Synthesis of Team Classification Literature," *Human Resource Development Review*, vol. 11, no. 1, pp. 97–129, 2012.

[16] Y. Loo, C. Gong, and M. Meghjani, "A Hierarchical Approach to Population Training for Human-AI Collaboration," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2023, pp. 3011–3019.

[17] H. Hu, A. Lerer, A. Peysakhovich, and J. N. Foerster, ""Other-Play" for Zero-Shot Coordination," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2020, pp. 4399–4410.

[18] X. Puig, E. Undersander, A. Szot, M. D. Cote, T.-Y. Yang, R. Partsey, R. Desai, A. W. Clegg, M. Hlavac, S. Y. Min, V. Vondruš, T. Gervet, V.-P. Berges, J. M. Turner, O. Maksymets, Z. Kira, M. Kalakrishnan, J. Malik, D. S. Chaplot, U. Jain, D. Batra, A. Rai, and R. Mottaghi, "Habitat 3.0: A Co-Habitat for Humans, Avatars and Robots," *arXiv preprint*, vol. arXiv:2310.13724, 2023.

[19] F. Ho, R. Geraldes, A. Gonçalves, B. Rigault, B. Sportich, D. Kubo, M. Cavazza, and H. Prendinger, "Decentralized Multi-Agent Path Finding for UAV Traffic Management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 997–1008, 2022.

[20] Y. Chen, J. Arkin, Y. Zhang, N. Roy, and C. Fan, "Scalable Multi-Robot Collaboration with Large Language Models: Centralized or Decentralized Systems?" *arXiv preprint*, vol. arXiv:2309.15943, 2023.

[21] R. Toro Icarte, T. Klassen, R. Valenzano, and S. McIlraith, "Using Reward Machines for High-Level Task Specification and Decomposition in Reinforcement Learning," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2018, p. 2107–2116.

[22] R. Toro Icarte, T. Q. Klassen, R. Valenzano, and S. A. McIlraith, "Reward Machines: Exploiting Reward Function Structure in Reinforcement Learning," *Journal of Artificial Intelligence Research*, vol. 73, pp. 173–208, 2022.

[23] A. E. Elsefy, "A task decomposition using (HDec-POSMDPs) approach for multi-robot exploration and fire searching," *International Journal of Robotics and Mechatronics*, vol. 7, no. 1, pp. 22–30, 2020.

[24] A. Preece, W. Webberley, D. Braines, E. G. Zaroukian, and J. Z. Bakdash, "SHERLOCK: Experimental evaluation of a conversational agent for mobile information tasks," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 6, pp. 1017–1028, 2017.

[25] S. Hunold and B. Przybylski, "Scheduling.jl - Collaborative and Reproducible Scheduling Research with Julia," *arXiv preprint*, vol. arXiv:2003.05217, 2020.

[26] D. Furelos-Blanco, M. Law, A. Jonsson, K. Broda, and A. Russo, "Induction and Exploitation of Subgoal Automata for Reinforcement Learning," *Journal of Artificial Intelligence Research*, vol. 70, pp. 1031–1116, 2021.

[27] D. Furelos-Blanco, M. Law, A. Jonsson, K. Broda, and A. Russo, "Hierarchies of Reward Machines," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2023, pp. 10 494–10 541.

[28] L. Ardon, D. Furelos-Blanco, and A. Russo, "Learning Reward Machines in Cooperative Multi-Agent Tasks," in *Proceedings of the Neuro-Symbolic AI for Agent and Multi-Agent Systems (NeSyMAS) Workshop at the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2023.