Learning Team Coordination to Traverse Adversarial Environments

Zechen Hu, Manshi Limbu, Daigo Shishika, Xuesu Xiao, and Xuan Wang

***Abstract*— This paper seeks to solve the coordination of a team of agents traversing a path in the presence of adversaries. Our goal is to minimize the overall cost of the team, which is determined by (i) accumulated risk when agents stay in adversary-impacted zones and (ii) mission completion time. During traversal, agents can reduce their speed and act as a 'guard' (the slower, the better), which can decrease the risks certain adversary incurs. This leads to a trade-off between agents' guarding behaviors and their travel speeds. The formulated problem is highly non-convex and cannot be efficiently solved by existing algorithms. We provide theoretical analysis regarding the team's coordination strategies when the number of agents and adversaries is small. As the scale of the problem expands, solving the optimal solution is challenging, therefore, we employ reinforcement learning techniques by developing new encoding and policy-generating methods. Simulations demonstrate that our learning methods can efficiently produce team coordination behaviors. We discuss the reasoning behind these behaviors and explain why they reduce the overall team cost.**

I. Introduction

Coordination of multi-agent systems has been studied under various contexts [1], including cooperative path planning [2], resource sharing and task allocation [3], and geometric formation maintenance [4]. Complementary to the challenges addressed in these works, in this paper, we introduce a new problem centered around adversarial risk management, which requires agents' cooperative dynamic behavioral decisions. Considering graph-based representation, such team coordination has been studied in [5], [6]. In this work, as shown in Fig. 1, we consider a team of agents traversing a continuous path with adversaries. Agents accumulate 'risks' when traveling zones controlled by adversaries, and such risks can be reduced by agents if they slow down and 'guard' a certain adversary. We define the team cost as a combination of mission completion time and accumulated risk. Therefore, minimizing this cost requires agents' coordination, trading off between their speed and adopting guarding behaviors. This adds a novel dimension of complexity and strategic decision-making, which is unattended in conventional multi-robot coordination tasks.

The scenario we've formulated has direct applications in the military domain. For instance, when multiple vehicles need to traverse enemy-controlled territories, suppressive fire from allies can mitigate threats posed by the enemy. The scenario also has applications in civilian contexts. For example, as multiple firefighters pass a fire-engulfed corridor, some members might deploy countermeasures to quell flames, ensuring safer passage for their peers. The formulated multi-agent coordination problem is challenging due to the combinatorial nature of agents' behaviors, their

George Mason University. {zhu3, klimbu2, dshishik, xiao, xwang64}@gmu.edu. Supported by Tactical Behaviors for Autonomous Maneuver (TBAM) collaborative research program (W911NF-22-2-0242).

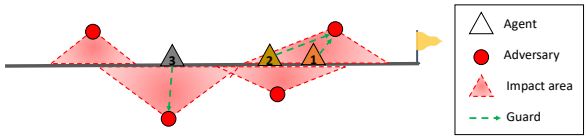


Fig. 1: A team of agents traversing a path with adversaries.

hybrid actions (speed and guard), and multiple constraints embedded through agent-adversary interactions.

Literature review. To address the formulated risk minimization problem with hybrid decision variables and constraints, feasible approaches exist in optimization literature such as Mixed Integer Programming (MIP). MIP has been applied to various multi-agent coordination problems, including task allocation [7], multi-robot path planning [8], environmental coverage and exploration [9]. Most existing MIP solvers, such as interior point [10] and branch and bound [11] methods, are sensitive to the number of variables. Consequently, the mentioned applications either have relatively small scales or can leverage decomposition techniques to break the problem into smaller sub-problems and reduce problem size. For example, in path planning, a vehicle plans its path individually and simply needs to ensure it doesn't obstruct other vehicles' paths [8]. However, in our work, the coordination of agents occurs at every time step, depending on their actions and states. Considering the whole trajectory can quickly drive traditional optimization techniques intractable. Additionally, our problem features non-smooth non-convex cost functions and constraints, which can make MIP solvers numerically unstable, or converge to sub-optimal solutions [12].

While optimization techniques suffer from computational complexity, Reinforcement Learning (RL) methods allow agents to employ trial and error to efficiently find empirical solutions for complex problems. For centralized problem solving, Deep Q-Networks (DQN) [13], a value-based RL method suitable for discrete actions, has been applied to learn team formations in battle games [14]. For complex tasks with continuous actions such as traffic optimization [15], Advantage Actor-Critic (A2C) methods [16] can achieve faster convergence and better exploration due to the use of a policy-based model as an actor. Building on A2C, there are generalizations such as Proximal Policy Optimization (PPO) [17] and Deep deterministic policy gradient (DDPG) [18] with improved stability or data efficiency. For systems with hybrid action space, there exist hybrid-PPO [19] methods that can output discrete actions simultaneously with continuous actions. In addition, considering the agent-based nature of our problem, the mentioned algorithms also have multi-agent variants such as Deep Coordination Graph [20], Multi-agent PPO [21], and Multi-Agent DDPG [22], allowing for decentralized execution, where each agent learns a local

model and determines actions according to local observation. The key advantage of MARL is to improve algorithmic scalability. However, the solution may be sub-optimal due to partial information.

Statement of contribution The contributions of this paper include (i) the formulation of a new multi-agent coordination problem with guard behavior between team members to reduce adversary risks; (ii) a theoretical analysis centered on the single adversary scenario; (iii) the deployment of a Hybrid Proximal Policy Optimization algorithm for our problem with special treatment of reward reshaping, as well as a unique multi-weighted hot encoding for representing agents' states; (iv) discussion of simulation results, elucidating the rationale behind observed behaviors and their efficacy in reducing the collective team cost.

II. Problem Formulation

In this section, we will first formulate the traversing *task* of the multi-agent team, then introduce the notions of *risk* and *guard*. Based on these, we quantify the *team cost* and describe our *problem of interest*. Throughout the following definitions, adversaries are considered to be heterogeneous while agents are homogeneous.

Task: Consider a number of n agents traversing a path of length L. The position of the i-th agent along the path at time t is represented by $s_i^t \in [0, L]$. Let $v_i^t \in [0, v_{\max}]$ denote the speed of the i-th traveling agent at time t. Thus, the position update for each traveling agent is given as

$$s_i^{t+1} = s_i^t + v_i^t \Delta t. \tag{1}$$

where Δt is the time interval. Before agent i arrives at the end of the path, each time-step will produce a time penalty, denoted by P_i^t. We define

$$P_i^t = \begin{cases} p & \text{if } s_i^t < L, \\ 0 & \text{if } s_i^t = L. \end{cases} \tag{2}$$

Risk: Let m denote the number of adversaries with each adversary located at $d_j \in [0, L]$. Suppose adversary j possesses a unique impact zone, denoted by $\mathcal{M}_j$, which incurs costs on traveling agents. At time step t, if an agent is in this region, i.e., $s_i^t \in \mathcal{M}_j$, a cost $r_{i,j}^t$ will be incurred.

$$r_{i,j}^t = \begin{cases} (\bar{r}_j - \eta_j |s_i^t - d_j|) & \text{if } s_i^t \in \mathcal{M}_j, \\ 0 & \text{if } s_i^t \notin \mathcal{M}_j, \end{cases} \tag{3}$$

where $r_{i,j}^t$ is bounded by $\bar{r}_j$ and decreases linearly with a co-efficient η_j as the distance between the agent and the adversary grows. We also assume $\mathcal{M}_j = [d_j - \ell, d_j + \ell] \bigcap [0, L]$, with $\ell = \frac{\bar{r}_j}{\eta_j}$, so that $r_{i,j}^t$ is always non-negative. Furthermore, $r_{i,j}^t = 0$ at the boundary of $\mathcal{M}_j$.

Guard: During traversal, if $s_k^t \in \mathcal{M}_j$, agent $k \in \{1, \cdots, n\}$ can counteract adversary j by reducing its speed and acting as a 'guard'. Specifically, let $g_k^t \in \{1, 2, \cdots, m\}$ denote the index of the adversary that agent k is guarding against at time t. Then, the risks that adversary j incurs to agents i, $\forall s_i^t \in \mathcal{M}_j$ are discounted to $\alpha_{k,j}^t r_{i,j}^t$, where

$$\alpha_{k,j}^t = \begin{cases} 1 - \beta \frac{(v_{\max} - v_k^t)}{v_{\max}} & \text{if } g_k^t = j, \\ 1 & \text{otherwise}, \end{cases} \tag{4}$$

is the discount factor with $\beta \in (0, 1)$. When $v_k^t = 0$, agent k achieves best guarding performance $\alpha_{k,j}^t = 1 - \beta$, while as $v_k^t \to v_{\max}$, the guarding effect vanishes. Furthermore, we assume the guarding effects stack with each other, thus, considering all agents in the system guarding an adversary j, the risk it incurs to agent i is discounted by all guards as $\prod_{k=1}^n \alpha_{k,j}^t r_{i,j}^t$. Note that the guarding behavior leads to agents' coordination, as one agent can decrease its speed to benefit all traveling agents within the influence region of the guarded adversary.

Team Cost: Let T be the total time for all agents in the team to traverse the path. Based on the above definitions, the team cost considers the risks and time penalties accumulated by all agents,

$$\boldsymbol{J} = \sum_{i=1}^n (\boldsymbol{R}_i + \boldsymbol{P}_i), \tag{5}$$

where $\boldsymbol{R}_i = \sum_{t=1}^T R_i^t \Delta t$ and $\boldsymbol{P}_i = \sum_{t=1}^T P_i^t \Delta t$. with

$$R_i^t = \sum_{j=1}^m \prod_{k=1}^n \alpha_{k,j}^t r_{i,j}^t. \tag{6}$$

Problem of Interest: In each time step, agent i's action is composed of traveling speed v_i^t and guard target g_i^t. To strategically design all agents' behaviors, let $\boldsymbol{v}^t = \{v_1^t, \ldots, v_n^t\}$, and $\boldsymbol{g}^t = \{g_1^t, \ldots, g_n^t\}$. The problem is to minimize the team cost, i.e., $t \in \{1, \cdots, T\}$

$$\min_{\{\boldsymbol{v}^t, \boldsymbol{g}^t\}} \boldsymbol{J}. \tag{7}$$

III. Method

In this section, we present methods for solving the formulated problem. We start by theoretically analyzing the optimal team strategy. However, due to the problem's complexity, the analysis is limited to a simplified scenario with only one adversary. For more complicated cases, we introduce proximal policy optimization (PPO) based RL algorithms with a special multi-weighted hot state encoding mechanism and reward reshaping to improve training efficiency.

A. Theoretical Analysis for the case of one adversary

If considering a single adversary, agents' guard action g_k^t becomes predetermined: all agents will guard the adversary when possible. Under this condition, the minimization of $\boldsymbol{J}$, narrows down to optimizing v_i^t of agents at every time step. We note that $\boldsymbol{J}$ is composed of $\sum_{i=1}^n \boldsymbol{R}_i$ and $\sum_{i=1}^n \boldsymbol{P}_i$. To minimize time penalty $\sum_{i=1}^n \boldsymbol{P}_i$, the strategy is trivial, where all agents should simply move at full speed. To minimize the accumulated risk $\sum_{i=1}^n \boldsymbol{R}_i$, agents' actions should follow a more complicated strategy, which is summarized in the following lemma.

Lemma 1. *Suppose $\Delta t \to 0$. $\sum_{i=1}^n \boldsymbol{R}_i$ is minimized if the team strategy is as follows: (i) Agents move across the adversary-impacted zone one-by-one while all other agents perform guard either at the start or the end of the zone; and (ii) The moving agent keeps maximum speed until arriving at the end of the adversary-impacted zone.*

Proof. The definition of R_i^t in (6) implies that the agent's risk per time step depends on the guard discount $\alpha_{k,j}^t$ and the risk $r_{i,j}^t$. From (3), we know $r_{i,j}^t = 0$ when $s_i = d_j \pm \ell$. This means agents incur no cost and can provide full risk discounts $\alpha = 1 - \beta$ if they stay and guard at the border of the impacted zone. Thus, to make full use of the guarding effect, the team must follow the strategy described in (i).

Now, suppose only agent i is moving. Then $\forall k \neq i$, $\alpha_{k,j}^t = 1 - \beta$ and $\boldsymbol{R}_i = \sum_{t=1}^{T}(1-\beta)^{n-1}\alpha_{i,j}^t r_{i,j}^t \Delta t$. The trade-off concerns whether agent i should slow down to reduce $\alpha_{i,j}^t$ or speed up to reduce the total travel time T. For ease of performing analysis, given $\Delta t \to 0$, we transition the risk accumulation to a continuous form, which reads[1]

$$\begin{aligned}\boldsymbol{R}_i &= (1-\beta)^{n-1}\int_0^T (1-\beta+\frac{\beta v_i}{v_{\max}})r_{i,j}dt \\ &= 2(1-\beta)^{n-1}\int_{\frac{T}{2}}^T (1-\beta+\frac{\beta v_i}{v_{\max}})(\bar{r}_j - \eta_j(s_i-d_j))dt\end{aligned}$$

where $v_i > 0$ and $s_i^{t=T} = \int_0^T v_i dt = 2\ell$. The second equation is obtained by utilizing the symmetric property of the adversary-controlled zone, where, for optimal strategy, we only compute the risk accumulated during the second half of the process, i.e., $\int_{\frac{T}{2}}^T v_i dt = \ell$. Now, by defining $y_i = s_i - d_j$ and utilizing $\frac{dy_i}{dt} = v_i$, one can substitute the integration variable from dt to dy_i, which reads,

$$\begin{aligned}\boldsymbol{R}_i =&\ 2(1-\beta)^{n-1}\int_0^{\ell}(1-\beta+\beta\frac{v_i}{v_{\max}})(\eta_j y_i)\frac{dy_i}{v_i} \\ =&\ 2(1-\beta)^{n-1}\int_0^{\ell}(\frac{1-\beta}{v_i}+\frac{\beta}{v_{\max}})(\eta_j y_i)dy_i.\end{aligned}$$

Clearly, $\boldsymbol{R}_i$ is minimized if $\forall t, v_i = v_{\max}$, which follows the strategy described in (ii). □

The result in Lemma 1 assumes $\Delta t \to 0$. In numerical simulation, the discretization may lead to small changes to agents' coordination. Nevertheless, in general, Lemma 1 describes a 'bang-bang behavior' [23]. By either completely stopping or moving at full speed, agents can maximize the guard effect and reduce team risk. However, this strategy is also time costly since only one agent moves at a time. To quantify the cost of crossing this risk zone, we have

$$\sum_{i=1}^n \boldsymbol{P}_i = \sum_{i=1}^n \int_{t=0}^T p dt = \frac{n^2 \ell p}{v_{\max}} \tag{8a}$$

$$\sum_{i=1}^n \boldsymbol{R}_i = n(1-\beta)^{n-1}\frac{\ell \bar{r}_j}{v_{\max}} \tag{8b}$$

where T grows linearly with n, thus, $\sum_{i=1}^n \boldsymbol{P}_i$ grows quadratically with n. $\sum_{i=1}^n \boldsymbol{R}_i$ is obtained from the last equation in the proof of Lemma 1, which decreases exponentially with n. Following (8), if $\bar{r}_j$ is large and n is small, $\sum_{i=1}^n \boldsymbol{R}_i$ will dominate the cost so that the team should follow the strategy in Lemma 1. However, as n grows, $\sum_{i=1}^n \boldsymbol{P}_i$ will eventually dominate the cost, necessitating a change of strategy into two possible variations: (i) Agents still either stop to guard or move at full speed, but some may start moving before other agents completely cross the risky zone. (ii) Several agents move with intermediate speeds that perform move and guard simultaneously. Both strategies are observed later in our simulations section, depending on the environment setup. Nevertheless, these behaviors consider multiple agents' movements in a coupled manner, which makes the theoretical analysis intractable. Motivated by this, in the following, we seek to use reinforcement learning methods to solve the problem.

[1]For continuous representation, the notation for time t is omitted.

B. MDP Formulation and RL methods

The Markov decision process (MDP) is a discrete time control process, defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \gamma, R)$, including state, action, state transition, discount factor, and reward. Here, we propose a centralized MDP to address the multi-agent coordination problem formulated in Sec. II. Specifically, our state space $\mathcal{S}$ represents all agent positions at each time step. Let $\boldsymbol{s}^t \in \mathcal{S}$ be the state set at time t, we have $\boldsymbol{s}^t = \{s_1^t, \ldots, s_n^t\}$, $s_i^t \in [0, L]$, and the state space is

$$\mathcal{S} \coloneqq [0, L] \times [0, L] \times \cdots \times [0, L] = [0, L]^n.$$

Correspondingly, for the actions of agents, we have $a_i^t = (v_i^t, g_i^t)$, which is a hybrid combination of continuous speed $v_i^t \in [0, v_{\max}]$ and discrete guard behavior $g_i^t \in \{1, 2, \ldots, m\}$. The action space for all agents is

$$\mathcal{A} \coloneqq \left([0, v_{\max}] \times \{1, 2, \ldots, m\}\right)^n. \tag{9}$$

Based on $\mathcal{S}$ and $\mathcal{A}$, the state transition $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ follows agents' motion dynamics (1), which is deterministic. $R(\boldsymbol{s}^t, \boldsymbol{a}^t)$ is the immediate reward of action $\boldsymbol{a}^t \in \mathcal{A}$ with state $\boldsymbol{s}^t \in \mathcal{S}$, defined as the negative team cost

$$R(\boldsymbol{s}^t, \boldsymbol{a}^t) \coloneqq -\sum_{i=1}^n (R_i^t + P_i^t). \tag{10}$$

We complete our MDP formulation by choosing a discounter factor $\gamma = 0.995$. The goal of RL is to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$ to maximize the expected cumulative reward for the whole team over the task horizon T, i.e.,

$$\max_{\pi} \quad \mathop{\mathbb{E}}_{\boldsymbol{a}^t \sim \pi(\cdot|\boldsymbol{s}^t)} \left[\sum_{t=0}^T (\gamma)^t R^t\right]. \tag{11}$$

Multi-weighted hot state encoding. To employ RL methods to solve our MDP problem, we can directly feed a vector representation of agents' states (positions) to the model. However, we observe that the dimension of our state space is much smaller than that of the action space due to the joint speed and guard behaviors. This discrepancy hinders the neural network's reasoning capabilities, which cannot efficiently learn parameters [24]. Inspired by the one-hot encoding, we seek to expand the state space. However, one-hot encoding is typically suited for discrete variables, whereas our state space is continuous. To address this, we introduce a new *weighted hot encoding* mechanism, which represents a continuous variable as the weighted average of two neighboring one-hot vectors. Specifically, let $h(x) \in [0, 1]^L$ denote the weighted hot encoding for a state $x \in$

$[0, L]$, and $x = x_{\text{int}} + x_{\text{dec}}$, which has both integer and decimal parts. Let $h(x)[k], k \in \mathbb{Z}$ denote the kth element of vector $h(x)$. Then $h(x)$ is a vector with two nonzero entries: $h(x)[x_{\text{int}}] = 1 - x_{\text{dec}}$ and $h(x)[x_{\text{int}} + 1] = x_{\text{dec}}$. As a simple example, if $L = 4$ and $x = 3.2$. Since $3.2 = 0.8 \times 3 + 0.2 \times 4$, one has, $h(x) = [0 \;\; 0 \;\; 0.8 \;\; 0.2]^\top$.

Since we consider a centralized MDP, given multiple agents, we vector stack their weighted hot encoding and obtain the Multi-weighted hot state encoding as follows:

$$\widetilde{\boldsymbol{s}}^t = \text{vec}\{h(s_1^t), h(s_2^t), \cdots, h(s_n^t)\} \in [0, 1]^{nL} \tag{12}$$

Here, we assume the task has proper length L. If L is too long, one can re-scale it to improve the traceability of $\widetilde{\boldsymbol{s}}^t$. Finally, it is worth mentioning that one-hot encoding is typically used for categorical variables. In our problem, the positions of agents, whether inside or outside of adversary-impacted zones, correspond to completely different properties and different feasible guard actions. This distinction favors one-hot encoding, as all input entries are orthogonal to each other. This fact further justifies why multi-weighted hot state encoding can enhance our learning efficiency.

Reward reshaping. Since our task requires all agents to move to the terminal position, it is common to introduce a one-time constant reward $Q(\boldsymbol{s}^t) = q$, if $\boldsymbol{s}^t = \{L, \cdots, L\}$; $Q(\boldsymbol{s}^t) = 0$, otherwise. However, this terminal reward is so sparse that it provides limited guidance to agents for state-space exploration and policy updates. Due to the greedy nature of the action section process, agents are reluctant to enter adversary zones. To address this, we further introduce a reshaping reward $F(\boldsymbol{s}^t, \boldsymbol{s}^{t+1}) = c \sum_{i=1}^{n} \left(\gamma s_i^{t+1} - s_i^t\right)$, which incites agents to move forward and speeds up the learning process [25]. The final reshaped reward reads

$$\widetilde{R}(\boldsymbol{s}^t, \boldsymbol{a}^t) = R(\boldsymbol{s}^t, \boldsymbol{a}^t) + Q(\boldsymbol{s}^t) + F(\boldsymbol{s}^t, \boldsymbol{s}^{t+1}) \tag{13}$$

where $\boldsymbol{s}^{t+1}$ is determined by $\boldsymbol{s}^t, \boldsymbol{a}^t$ through $\mathcal{T}$. We note that the reshaped reword does not change the optimal solution compared to the original formulation. This is guaranteed by [26], as both $Q(\boldsymbol{s}^t)$ and $F(\boldsymbol{s}^t, \boldsymbol{s}^{t+1})$ can be rewritten into a potential-based function: $\gamma\Phi(\boldsymbol{s}^{t+1}) - \Phi(\boldsymbol{s}^t)$, where $\Phi(\cdot)$ is a real-valued function of state and γ is the discount factor.

RL Implementation Combining the formulated MPD with multi-weighted hot state encoding and reward reshaping, we use two proximal policy optimization (PPO) based RL algorithms to solve the multi-agent path traveling problem. The key difference lies in the way we handle the hybrid action space. First, for simplicity, we consider pure discrete action space, assuming agents only take integral speeds. This has led to a standard PPO with discrete actions (D-PPO) [27], and the proposed multi-weighted hot state encoding degrades to multi-one hot encoding. Second, we consider PPO with hybrid action space (H-PPO), and let the actor-network simultaneously output continuous speed actions and discrete guard actions. The policy losses (*log* probabilities) of the two actions are combined and used for training the network parameters. A conceptual diagram of the RL implementation is shown in Fig. 2, with a centralized structure to handle all agents' rewards and actions. Leaving as our future work, a possible decentralized implementation of the RL paradigm is to let each agent possess a local model of Fig. 2. Then, leveraging our multi-weighted-hot encoder, if the agent cannot observe certain agent's states, the corresponding weighted-hot vector has all entries being zero.

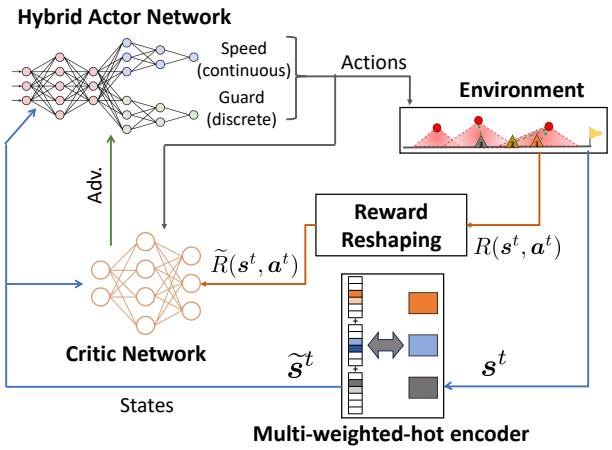


Fig. 2: RL Implementation: H-PPO with multi-weighted hot encoding and reward reshaping

IV. RESULTS

In this section, we present simulation results to validate the statements in theoretical analysis and to demonstrate that the introduced learning methods can efficiently produce team coordination behaviors. For more complex cases, we comprehensively discuss the reasoning behind these behaviors and why they reduce the overall team cost.

A. Simulation Environment

We consider four different environments as visualized in Fig. 3, including one adversary (M1), two adjacent adversaries (M2a), two adversaries with an overlap (M2b), and three adversaries with multiple overlaps (M3). We choose the following environment parameters: time interval $\Delta t = 1$, max agent speed $v_{\max} = 3$, risk co-efficient $\eta = 1$, time penalty $p = 1$, guard discount co-efficient $\beta = 0.6$, terminal state reward $q = 10$, and reshape reward coefficient $c = 0.2$. All rewards, before sending to D-PPO, H-PPO models, are re-scaled by $\frac{1}{20}$ for normalization purposes.

To visualize results, in Figs. 4 and 5, we use the x-axis to represent time and the y-axis to represent agent positions in the environment. Thus, the slope represents the agent's speed. The color dots on the trajectories represent the adversary

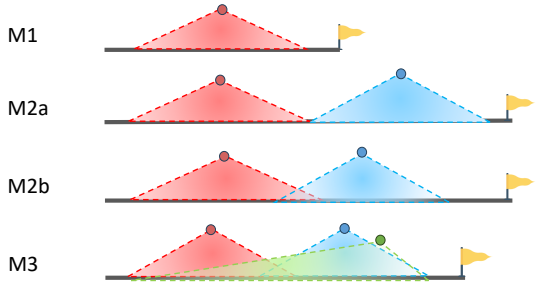


Fig. 3: Experiment environments with different adversary configurations. The height represents the unit risk each adversary generates at different locations.

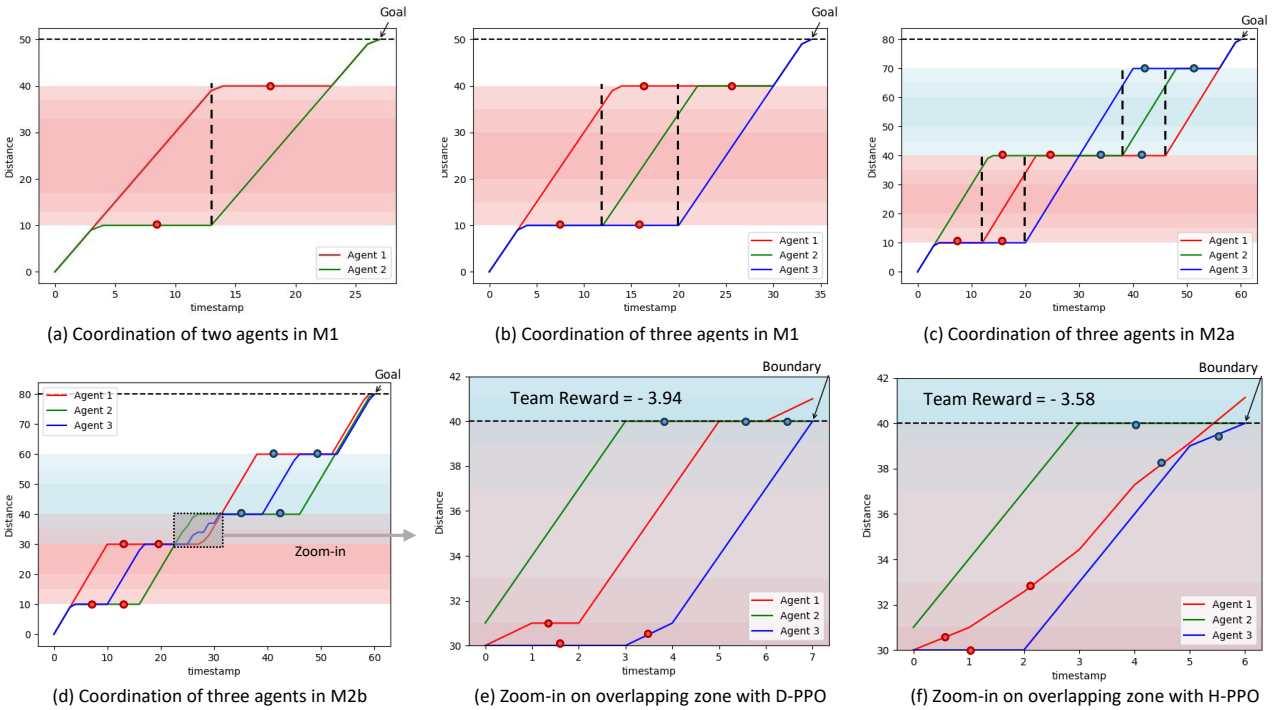


Fig. 4: Using D-PPO and H-PPO to solve team coordination problem with first three environments in Fig. 3.

the agent is currently guarding against. The shades are the adversary-impacted zones with darker colors in the middle to represent higher risk, corresponding to Fig. 3.

B. Validating the performance of learning with Lemma 1.

We discuss the alignment of results in Fig. 4a-c with Lemma 1. The D-PPO and H-PPO methods generate almost the same results except for slight differences in velocity when agents leave or approach the boundaries of adversary-impacted zones. These differences are caused by speed discretization and lead to minor changes to the final reward. For conciseness, we only visualize the results from D-PPO. In Fig. 4a, the two agents' behaviors follow exactly as the statements in Lemma 1. One agent guards at the edge until the other agent arrives at the other end of the adversary-impacted zone at full speed. Then, the two agents switch their roles to cooperatively accomplish the task with minimum cost. In Fig. 4b, as the number of agents increases, we observe (from the vertical dashes) a change in agents' coordination such that the guarding agents will start moving 2 seconds before the traveling agents arrive at the other end. This is in line with our hypothesis-(i) at the end of Sec. III-A: due to the increase of agent's number, the impact of time penalty in (8a) grows while the impact of risk accumulation in (8b) decreases. Finally, Fig. 4c demonstrates the effectiveness of the RL methods for solving coordination algorithms over a longer distance. Since the two adversaries do not have overlaps, the three agents simply reproduce coordination behaviors in Fig. 4b over the two zones, respectively.

C. Team coordination under complex environments

We employ D-PPO and H-PPO methods to solve optimal coordination under M2b and M3 environments, where theoretical analysis becomes challenging. For the case of two adversaries with an overlap (M2b), the results of D-PPO and H-PPO show relatively consistent strategies for the $[10, 30]$ and $[40, 60]$ zones, as shown in Fig. 4d, but have variant behaviors between the overlapping zone $[30, 40]$. An oscillation in the training loss with D-PPO is also observed. One potential explanation for this oscillation in loss is the homogeneity of the agents in our setup. This homogeneity can result in multiple equivalent yet distinct optimal strategies (by switching the orders of agents), that are scattered across the state-action space. If these distinct optimal solutions appear randomly in training data, the parameter can be diverted, leading to a spike in loss. To address this, we first assume the behaviors on lateral sides are optimal, then zoom in to perform a new analysis only on this overlapping zone. We train the model with smaller learning rates and the results are shown in Fig. 4e-f. Note that agents start at $\{30, 31, 30\}$ instead of $\{30, 30, 30\}$ because when entering the zone, agent 2 moves with $v_{\max} = 3$ from $s_2 = 28$ and directly arrives at $s = 31$. For a similar reason, the terminal states are $\{41, 40, 40\}$. In both Fig. 4e and f, at least one agent takes intermediate speeds that perform move and guard behaviors simultaneously. This aligns with the hypothesis-(ii) at the end of Sec. III-A. When agents are close to the boundaries of the overlapping zone, their risks are dominated by red and blue adversaries, respectively. As observed in plots, the stationary agent always guards the adversary which causes more risk, while the agent with intermediate speed will guard the other adversary. Moreover, a comparison of the results reveals that H-PPO, with its ability to handle continuous speeds, can refine strategies more effectively, resulting in better rewards. The asymmetry in agent's behavior may be due to time discretization, where the cost is computed at the

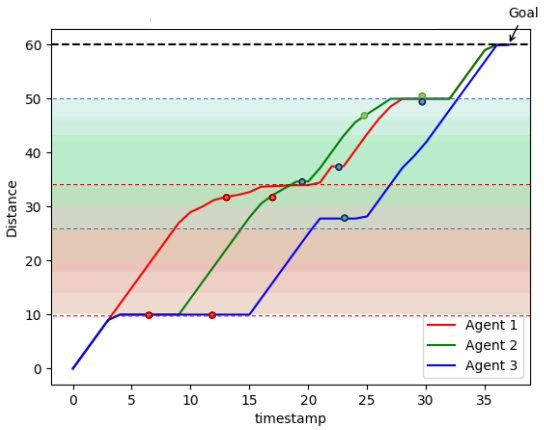
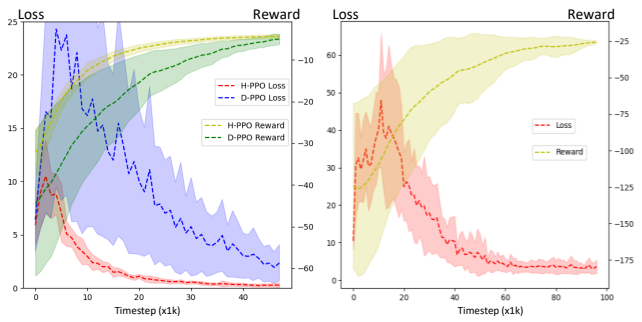


Fig. 5: Coordination of three agents in M3 using H-PPO.



(a) Training the overlapping zone in M2b using H-PPO and D-PPO.

(b) Training M3 environment with three agents using H-PPO.

Fig. 6: Convergence results for training loss and rewards.

end of each time step. By bringing the obtained strategy for the zoom-in zone back to the entire trajectory, the combined trajectory leads to better reward than using D-PPO and H-PPO to solve the full-scale problem directly.

For the case of three adversaries with multiple overlaps (M3), we deploy three agents and obtain coordination results as shown in Fig. 5. We added colored dashes to better visualize the boundaries of red and blue adversaries. The environment's complexity makes it difficult to judge the optimality of the obtained coordination. In the following, we only discuss the reasoning behind the obtained result and explain why it reduces the overall team cost. First, according to Fig. 3, the M3 environment can be viewed as a modified version of M2b with an extra green adversary. This green adversary generates an unsymmetric risk zone, which poses little risk early in the path but grows large as agents proceed. Consequently, in Fig. 5, agents first follow a pattern very similar to that of Fig. 4d. However, midway through the path, as risks associated with the green adversary become large, the predecessor agent 1 does not fully stop to guard others. Instead, it takes an intermediate speed to guard the red adversary. This lasts until agent 1 meets agent 2 and both leave the boundary of the red adversary. On the other hand, we observe agent 3 stops in the middle of the path to perform guard. This happens because, at the moment, the other two agents are suffering huge risks from both blue and green adversaries. By stopping and thereby increasing its own risk, agent 3 contributes to the cost-saving of the whole team. Finally, when agent 1 and 2 both arrives at the boundary, they help agent 3 by guarding red and blue adversaries, respectively. We note that the coordination presented in Fig. 5 may not be the global optimal, and its optimality gap is difficult to quantify. However, as discussed above, the observed agent coordination does exhibit rational behaviors, with the goal of reducing the overall team cost.

D. A Naive Baseline and Reward Comparison

Although determining optimal team strategies is challenging, we aim to quantify the effectiveness of the learned coordination by introducing a naive baseline strategy. In this strategy, we assume all agents always move at the same speed and each agent individually uses greedy choice for guarding adversaries. This simplification decouples all agents' actions, facilitating the exhaustive determination of an action sequence that maximizes the reward. For all environments listed in Fig. 3, we deploy three agents. The team rewards by employing the naive baseline, D-PPO, and H-PPO are given in Table I[2]. Finally, to visualize the training process, in Fig. 6, we use the zoom-in zone of the M2b environment and the complete M3 environment as representative results to show training losses and rewards. From both comparisons, we observe that H-PPO performs better than the other methods. D-PPO struggles to converge for M3 environment.

TABLE I: Rewards Comparison

Environments	**Baseline**	**D-PPO**	**H-PPO**
M1	-6.94	-5.80 ± 0.2	-5.78 ± 0.0
M2a	-20.18	-8.63 ± 0.7	-8.22 ± 0.4
M2b	-35.66	-24.76 ± 1.7	-22.10 ± 0.8
M3	-44.70	-40.13 ± 8.5	-26.87 ± 2.0

V. Conclusion and Future Work

We have formulated a coordination problem considering a team of agents traversing a path with adversaries. The cost of the team was mainly determined by the time penalty and the risk they accumulated when crossing adversary-impacted zones, which can be reduced by agents' guard behavior when moving at a lower speed. In addition to the formulation of this problem, our contributions included a theoretical analysis of optimal coordination strategy for a single adversary scenario, as well as the implementation of an H-PPO method with reward reshaping and a new multi-weighted hot state encoding mechanism. We performed simulated experiments to validate the correctness of the theoretical analysis and the effectiveness of the H-PPO method. Based on the simulation results, we discussed the reasoning behind these behaviors in terms of reducing the overall team cost. Future work will consider developing a decentralized learning paradigm to achieve scalable coordination with a larger number of agents and more complicated environments. We also seek to expand the proposed formulation to 2-D environments considering the geometries of risk zones and terrain.

[2]Although added a new adversary, the reward of M3 is not significantly smaller than M2b because the environment length is shorter.

REFERENCES

[1] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *Ieee Access*, vol. 6, pp. 28573–28593, 2018.

[2] A. Torreno, E. Onaindia, A. Komenda, and M. Štolba, "Cooperative multi-agent planning: A survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–32, 2017.

[3] M. Afrin, J. Jin, A. Rahman, A. Rahman, J. Wan, and E. Hossain, "Resource allocation and service provisioning in multi-agent cloud robotics: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 842–870, 2021.

[4] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[5] C. A. Dimmig, K. C. Wolfe, and J. Moore, "Multi-robot planning on dynamic topological graphs using mixed-integer programming," *arXiv preprint arXiv:2303.11966*, 2023.

[6] M. Limbu, Z. Hu, S. Oughourli, X. Wang, X. Xiao, and D. Shishika, "Team coordination on graphs with state-dependent edge cost," in *20230 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2023.

[7] C. Zhang and J. A. Shah, "Co-optimizating multi-agent placement with task assignment and scheduling.," in *IJCAI*, pp. 3308–3314, 2016.

[8] J. Berger and N. Lo, "An innovative multi-agent search-and-rescue path planning approach," *Computers & Operations Research*, vol. 53, pp. 24–31, 2015.

[9] F. Gul, A. Mir, I. Mir, S. Mir, T. U. Islaam, L. Abualigah, and A. Forestiero, "A centralized strategy for multi-agent exploration," *IEEE Access*, vol. 10, pp. 126871–126884, 2022.

[10] F. A. Potra and S. J. Wright, "Interior-point methods," *Journal of computational and applied mathematics*, vol. 124, no. 1-2, pp. 281–302, 2000.

[11] L. D. Beal, D. C. Hill, R. A. Martin, and J. D. Hedengren, "Gekko optimization suite," *Processes*, vol. 6, no. 8, p. 106, 2018.

[12] E. Klotz, "Identification, assessment, and correction of ill-conditioning and numerical instability in linear and integer programs," in *Bridging Data and Decisions*, pp. 54–108, INFORMS, 2014.

[13] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, "Deep exploration via bootstrapped dqn," *Advances in neural information processing systems*, vol. 29, 2016.

[14] E. A. O. Diallo and T. Sugawara, "Learning strategic group formation for coordinated behavior in adversarial multi-agent with double dqn," in *PRIMA 2018: Principles and Practice of Multi-Agent Systems: 21st International Conference, Tokyo, Japan, October 29-November 2, 2018, Proceedings 21*, pp. 458–466, Springer, 2018.

[15] H. Zhang, W. Chen, Z. Huang, M. Li, Y. Yang, W. Zhang, and J. Wang, "Bi-level actor-critic for multi-agent coordination," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 7325–7332, 2020.

[16] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," *Advances in neural information processing systems*, vol. 12, 1999.

[17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[18] S. Li, Y. Wu, X. Cui, H. Dong, F. Fang, and S. Russell, "Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 4213–4220, 2019.

[19] Z. Fan, R. Su, W. Zhang, and Y. Yu, "Hybrid actor-critic reinforcement learning in parameterized action space," *arXiv preprint arXiv:1903.01344*, 2019.

[20] W. Böhmer, V. Kurin, and S. Whiteson, "Deep coordination graphs," in *International Conference on Machine Learning*, pp. 980–991, PMLR, 2020.

[21] J. Wang and L. Sun, "Dynamic holding control to avoid bus bunching: A multi-agent deep reinforcement learning framework," *Transportation Research Part C: Emerging Technologies*, vol. 116, p. 102661, 2020.

[22] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[23] R. Bellman, I. Glicksberg, and O. Gross, "On the "bang-bang" control problem," *Quarterly of Applied Mathematics*, vol. 14, no. 1, pp. 11–18, 1956.

[24] M. Botvinick, S. Ritter, J. X. Wang, Z. Kurth-Nelson, C. Blundell, and D. Hassabis, "Reinforcement learning, fast and slow," *Trends in cognitive sciences*, vol. 23, no. 5, pp. 408–422, 2019.

[25] S. M. Devlin and D. Kudenko, "Dynamic potential-based reward shaping," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012), pp. 433–440, IFAAMAS, 2012.

[26] A. Y. Ng, D. Harada, and S. J. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proceedings of the 16th International Conference on Machine Learning*, pp. 278–287, 1999.

[27] C. C.-Y. Hsu, C. Mendler-Dünner, and M. Hardt, "Revisiting design choices in proximal policy optimization," *arXiv preprint arXiv:2009.10897*, 2020.