Experience-Based Action Advising for Multi-Agent Teaming

Shuyang Shi

CMU-RI-TR-25-11

April 15th, 2025

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Katia Sycara, *chair*
Jiaoyang Li
Renos Zabounidis

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

Abstract

In this work, we study how to improve coordination efficiency for multi-agent teams with heterogeneously experienced agents. In such a setting, experienced agents can transfer their knowledge to less experienced agents to accelerate their learning, while leveraging the students' initial expertise to inform what knowledge to transfer. Inspired by this idea, this work specifically assumes one teacher agent in the team, and explores how it can efficiently utilize these knowledge priors to effectively improve the students' training by performing experience-based action advising tailored to each student agent. We propose a novel teaching approach that leverages the teacher's policy to identify a student's pre-existing skill and subsequently assigns appropriate sub-tasks to each student based on a bandit formulation. As a result, student teammates are assigned to and advised through sub-tasks that enable them to leverage their skills and thus improve overall task convergence. We empirically evaluate our method on Grid World tasks with two and three agents, and in an Overcooked-AI task with three agents. Our method outperforms existing teacher-student approaches that do not consider prior knowledge, and achieves faster convergence than teaming without knowledge transfer, demonstrating that tailored action advising accelerates team learning and improves overall performance, particularly as student agents may have accrued prior experience in particular sub-tasks.

Acknowledgments

I would like to express my deepest gratitude to my advisor, Dr. Katia Sycara, for her invaluable support throughout my Master's studies. Her insight into research questions, combined with her expertise and experience in problem-solving, has served as a guiding beacon for me. I am also sincerely thankful to my committee members, Dr. Jiaoyang Li and Renos Zabounidis, for their insightful comments and constructive feedback and comments on my work.

I wish to thank Dr. Joseph Campbell, Dr. Simon Stepputtis, Dr. Woojun Kim, and Dr. Sophie Yue Guo for the guidance and assistance they generously provided. Their help at every stage of this project contributed significantly to my progress, and I am deeply appreciative of their expertise. I am also grateful to my labmates and friends – Muhan Lin, Weihao Zeng, Zifu Wan, Silong Yong, Benji Li, Nate Ludlow, and He Jiang... – thank you for all the conversations, laughter, and constant inspiration; you made this journey both memorable and enriching.

Finally, but most importantly, I owe heartfelt thanks to my girlfriend and my family for their unwavering love and support. Their belief in me has always been my greatest source of strength, and I am profoundly thankful for their encouragement.

Funding

This project is sponsored by the Honda Research funding: award number: 58629.1.1012949.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Efficient coordination of agents on the fly [41] is essential in domains such as disaster response, where rapid team formation can dramatically improve the effectiveness of the overall operation [21]. Assigning agents to reasonable roles [28, 44] based on their capabilities is effective for these problems, where efficiency can be improved by having agents specializing in small parts of the tasks instead of trying to be generalists. Past literature has studied the problem with the assumption that agents have prior experience on a par with each other. For example, in multi-agent reinforcement learning (MARL), a set of novice agents jointly learns to accomplish a task collaboratively. Role assignment and emerging methods [44, 45] are used to facilitate efficient learning based on modeling agents' latent states from their behaviors. While in the ad hoc teamwork literature, agents are assumed to be able to contribute to the task individually [5]. The ad hoc agent identifies the roles of the teammates [31] and adapts its behavior to fit into the team [37, 38].

However, the possibility of disparity in agents' prior experience is largely ignored, which is commonly seen in real-world scenarios. Think about three agents collaborating to save victims from a collapsed house. They need to clear up the debris obstacles, secure damaged structures and conduct medical support to the victims. It is possible that one of them is an experienced firefighter who knows a lot about all the work required, while the other two are random neighbors who come to help. In this case, the firefighter possesses sufficient knowledge for the task already, while other two teammates require additional learning to develop adequate skills. Addressing the

knowledge disparity [3] among them is essential for their effective collaboration.

The scenario above highlights the potential advantages of sharing knowledge through inter-agent transfer learning schemes [10], In this approach, the firefighter can guide and expedite the neighbors' learning [43] of essential skills, such as first aid and debris removal. In this context, because neighbors may differ in their prior experience, providing guidance that aligns each individual's expertise with the demands of the task is important for effective knowledge transfer [20] that sufficiently reuses agents' existing knowledge and avoids skill forgetting [33]. For instance, a neighbor with engineering expertise would be more effective clearing debris, while another with nursing experience could handle first aid. Assigning roles in this way allows the firefighter to focus on specialized tasks, such as securing damaged structures, while boosting neighbors to refine their skills most suitable to the task, ultimately improving team efficiency.

Building on this insight, we focus on leveraging skill identification to enhance the learning efficiency of less experienced agents during collaboration. We study the following question: *How can an experienced agent (firefighter) transfer knowledge to learner teammates (neighbors) with heterogeneous prior experience to improve learning efficiency?*

Specifically, we consider an ad hoc collaboration setting [41] in which agents have limited pre-coordination, making the problem of knowledge disparity more pronounced. Communication among agents is restricted, and there is no access to teammates' decision-making strategies [32]. We assume the experienced agent–illustrated in the example–can proficiently perform all required roles, while the learner teammates possess only limited task skills acquired from their previous experience. All agents share the same state and action spaces, as well as a common communication protocol for transmitting this information. Drawing inspiration from the scenario above, we propose a novel method to improve learning efficiency in such mixed-experience teams through effective knowledge transfer and skill reuse,driven by role assignment based on skill identification. We adopt action advising methods [43] for knowledge transfer because they operate with minimal communication and remain agnostic to the learners' internal learning algorithms.

Figure 1.1 shows an overview of our task in which an experienced agent (red) assesses the existing capabilities of two student agents (blue and green), assigns

suitable sub-tasks to them, and finally advises them on how to complete their respective tasks in order to improve the overall team reward. Our key contributions involve a two-stage framework as follows:

- **Task/Role Allocation based on Prior Experience** - Using a multi-armed bandit algorithm, the experienced teacher agent evaluates the skills of its lesser-skilled teammates and assigns appropriate sub-tasks.
- **Tailored Action Advising** - A policy advising algorithm that provides suitable guidance to accelerate the team's learning and overall task efficiency.

We empirically evaluate the proposed method on Grid World tasks and the Overcooked-AI task [7]. We also provide extensive ablation studies to verify that each individual component of the proposed method is necessary.

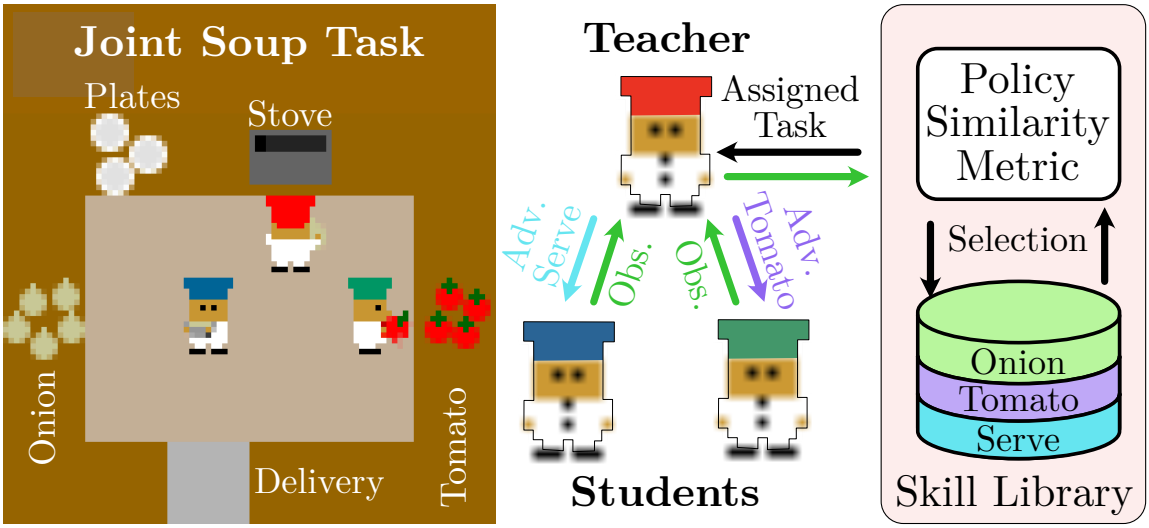

Figure 1.1: Overview of our method: The teacher (red) observes the heterogeneous students (green and blue) to determine which sub-task to teach each student while best utilizing their respective prior skills. We test our approach in the Overcooked-AI environment.

Chapter 2

Background

2.1 Related Works

Inter-agent transfer learning is an effective scheme to reduce sample complexity in MARL [12], where agents respond to instructions from others to improve their learning speeds. A typical form of instruction is action advice [43] which is also known as the teacher-student paradigm [14], where suggested actions are generated from the model of more competent teacher agents for student agents. The method is advantageous because it requires minimum assumptions on agents' common understanding of the system, i.e., actions and state observations [12], while other methods would require common internal states such as policy model structures [9].

Determining advice from which agent to use is essential for effective teaching in MARL systems. With the intuition that advice should come from more experienced agents, [2, 11, 20] model agents' history trajectories into confidence in states, which is higher when the agent visits the state more frequently. As a result, advice from agents with higher confidence is given to the agent who requires instruction. while good instructions are important for agents to learn, bad action advice is also crucial for adequate exploration [2, 23], especially in the early stages of learning. Therefore, instead of generating advice based on confidence heuristics, [29, 35, 52] learn teaching policies for agents, with objectives focusing on improving agents' learning performance, such as rewards [52] or value improvement [29, 35]. However, most of the literature assumes that agents learn together from scratch or with experience heterogeneity

only in parts of the states explored [20]. The problem of knowledge disparity among agents are not explicitly considered.

Role assignment and task decomposition are also important to improve the efficiency and scalability of MARL, which significantly reduces the policy search complexity from the joint policy space to individual roles and sub-tasks [16, 45]. Emergent roles and sub-tasks are commonly seen in MARL literature, where roles and sub-tasks are represented as features encoded from agents' history trajectories [44, 45, 51], policies are then learned for the group of agents that share similar roles or sub-tasks. These methods improve team learning efficiency in complex domains via policy sharing in similar agents and flexible role assignment dynamically constructed from experience. However, these constructive role features are not transferrable between agents. Pre-determined sub-tasks on the other hand, though difficult to craft in complex domains [50], are advantageous for knowledge reuse and transfer in MARL. [42] studies decomposing task-independent sub-tasks that can be reused in other tasks with minimal adaptation. And expert-provided decomposition of tasks can also be used as curriculums to pre-train agents for learning long-sequence coordination [16]. Furthermore, policies of the expert-decomposed sub-tasks can be specifically useful for transfer learning in MARL [19], which reduce the exploratory complexity significantly by providing advice which exactly how to collaborate.

Ad-hoc teamwork studies the problem of collaboration with unknown agents to achieve a shared goal [5, 41], where the ad hoc agent is supposed to join teammates with whom he/she has no or limited previous experience. In such a scenario, communication is usually restricted, as the agents involved are usually developed independently [32] and share minimum overlapping on world understanding and communication protocols [39]. Therefore, agent modeling [17] on teammates' roles and behaviors [31] is a priori condition for effective collaboration.

To tackle the problem, Bayesian framework is typically exploited [1, 4, 37, 38], where a posterior distribution over teammates' types [1] is estimated from the history observations on teammates' behavior [37]. The ad hoc agent usually holds a library of task models [5, 38] or behavior policies [4] to deal with every type of teammates. Although various approximation and reweighing methods are used, the numerical instability and the sensitivity to noise of Bayesian estimation make the method less competitive in complex domains. Feature-based teammate modeling

frameworks [8, 17, 34] are employed where a function approximator is learned from previous experience to map the trajectories of the observed new teammates to certain latent features. The behavior of the ad hoc agent is then conditioned on the features, which is updated [34] or zero-shot transferred [17] to collaborate with the new teammates. These methods hold the merit of generalizability and estimation stability, with the cost of a sophisticated training process and the requirement of a tremendous amount of data.

Though the domain shares similarity with our problem on the necessity of modeling and identification on agents' skills, the teammates in ad-hoc teamwork are assumed to be capable for the task [41]. The existence of novice and learner teammates is less investigated.

2.2 Preliminaries

Multi-Agent Reinforcement Learning We consider a fully cooperative Markov Game [30], which extends the Markov Decision Process (MDP) to multi-agent settings, where multiple agents interact within a shared environment and share the same reward function. A fully cooperative Markov Game is defined by the tuple $G = (N, S, \{A_i\}_{i=1}^N, P, R, \gamma)$, where N represents the set of agents, S is the set of global states, $\{A_i\}$ denotes the action spaces for each agent, and $P(s'|s, a_1, ..., a_N)$ is the transition function that defines the probability of transitioning from state s to state s' based on the joint actions. Here, all agents share the same reward function $R(s, a_1, a_2, ..., a_N)$, which maps the joint state-action pair to a shared reward. The goal is to learn policies $\pi = \{\pi_i\}_{i=1}^N$ that work together to maximize the cumulative discounted shared reward objective:

$$J(G, \pi) = \mathbb{E}_{s_{\tau+1} \sim P(\cdot|s_\tau, a_\tau), a_\tau \sim \pi(\cdot|s_\tau)} \sum_{\tau=0}^{\infty} \gamma^\tau R(s_\tau, a_\tau) \tag{2.1}$$

where γ is the discount factor, τ is the step variable in discrete setting, and a represents the joint action $\{a_1, \cdots, a_N\}$. To achieve this goal, numerous MARL algorithms have been proposed [13, 15, 24, 25, 36, 53]. In this paper, we consider a simple MARL algorithm, Independent PPO (IPPO) [13], which has been shown to be empirically

effective.

IPPO is a naive multi-agent variant of PPO where each agent has its own policy and value function. Specifically, each agent estimates its own advantage using Generalized Advantage Estimation [40] and then uses it to train its policy to maximize the advantage. Due to its simplicity and effectiveness, we use IPPO as a subroutine.

Action advising is a prominent knowledge transfer method in multi-agent reinforcement learning [12] where a teacher agent helps student agents learn faster on a target task by providing action advice during training. It reduces the burden of exploration by guiding the students to sample high-quality trajectories and can be especially successful in RL tasks with sparse rewards. Consider a teacher has a pre-trained policy π^T and a student is training its policy π^S. At time step τ, the student explores based on a behavior policy defined as

$$\tilde{\pi}^S(\cdot \mid s_\tau) = I \cdot \pi^T(\cdot \mid s_\tau) + (1 - I) \cdot \pi^S(\cdot \mid s_\tau) \tag{2.2}$$

where I is an $\{0, 1\}$ indicator which takes value 1 when advice is issued, and 0 otherwise. In action advising, it is usually assumed that the student agent always follows the given advice [6].

Multi-Armed Bandit (MAB) is an iterative decision-making framework that addresses the trade-off between exploration and exploitation to maximize cumulative rewards over time [26]. The problem involves K arms, each with an unknown value distribution $< D_1, \cdots, D_K >$. At each step τ , the agent selects an arm $j(\tau)$ and receives a reward $r(\tau) \sim D_{j(\tau)}$. The objective is to maximize the total reward over C rounds, equivalently minimizing the regret $R_C = C\mu^* - \sum_{\tau=1}^{C} r(\tau)$, where μ^* is the expected reward of the optimal arm. The Upper Confidence Bound (UCB) algorithm is a popular solution that selects the arm maximizing $\mu_{j(\tau)} + \sqrt{\frac{2 \ln \tau}{n_{j(\tau)}}}$, where $\mu_{j(\tau)}$ is the estimated mean reward and $n_{j(\tau)}$ is the number of times arm $j(\tau)$ has been chosen. UCB effectively balances exploring less tried arms and exploiting those with higher estimated rewards, facilitating both the discovery of optimal arms and the utilization of known high-reward options.

Chapter 3

Methodology

As discussed in Chapter 1, we aim to address the following question: *How can an experienced agent transfer knowledge to learner teammates with heterogeneous prior experience to improve the learning efficiency?* To answer this question, we first define our problem formulation and assumptions on the task, teacher agent, and student agents in Sec. 3.1. Then we propose a two-phase method where sub-task assignment is performed based on skill-identification in phase 1 (Sec. 3.2), and student agents are advised accordingly to refine their existing skills toward optimal collaboration in phase 2 (Sec. 3.3).

3.1 Problem Formulation

The multi-agent task T is defined as a fully cooperative Markov game of n agents $T = (N, S, \{A_i\}_{i=1}^n, P, R, \gamma)$ where $|N| = n$. The state $s \in S$ is fully observable, and the agents share the same action space, i.e., $A_i = A_j = A, \forall i, j \in N$. Assume T can be decomposed into n sub-tasks in a sub-task space $\{T^1, \cdots, T^n\} = \mathcal{T}$, each corresponding to a single agent MDP $T^j = (S, A, R^j, \gamma)$ with individual reward $R^j(s, a)$. Notice that R^i is usually conditioned on other agents behaviors in collaborative tasks. n agents can effectively solve T by each claiming a $T^j \in \mathcal{T}$ without repetition.

The experienced agent (Teacher) is assumed being pre-trained with R^j for sub-tasks $T^j \in \mathcal{T}$, thus acquired a policy library $\{\pi_*^1, \cdots, \pi_*^n\} = \Pi_*$ where each $\pi_*^j(a \mid s), S \times A \mapsto [0, 1]$ is an optimal solution to T^j, i.e., $\pi_*^j = \arg\max_\pi J(T^j, \pi)$.

A learner teammate (Students) has a policy $\pi_i(a \mid s), S \times A \mapsto [0, 1]$ drawn from an arbitrary policy distribution such that π_i is sub-optimal to any decomposed sub-tasks. We assume $n-1$ such students collaborating with the teacher agent for T, and denote the set of student agents as N_{-1}, representing the complement of teacher agent in task T. Since students are not assumed to be trained on tasks in $\mathcal{T}$, they can only learn from joint team reward R when teaming with the teacher agent without access to R^j for individual sub-tasks.

Our problem is to have the teacher agent identify on which subtask T^j could each student agent $i \in N_{-1}$ learn the most efficiently for the overall team objective $J(T)$, and transfer π^j_* to steer i's learning for T^j under the team reward R to optimize the performance.

We propose a two-phase algorithm in which the teacher first assigns sub-tasks to student agents based on an evaluation of their skills using MAB—the *sub-task assignment phase*, where a mapping from agent $i \in N$ to sub-task $T^j \in \mathcal{T}$ is determined. Notice that the sub-task for the teacher agent is also assigned, which the teacher agent executes using the corresponding optimal policy. Then the teacher teaches accordingly to refine students' abilities toward optimal collaboration with action advising methods—the *advising phase*. Specifically, the student agents' behavior is observed and compared with the teacher's sub-task knowledge under various sub-task assignment situations based on a policy similarity metric. The metric further guides teacher's MAB-based decision-making on what sub-task knowledge should be advised to each student for optimal team learning efficiency, as shown in Fig. 1.1.

3.2 Phase 1: sub-task Assignment

This phase aims to assign the appropriate sub-tasks to the student agents by evaluating their pre-existing skills. To achieve this, we propose using a policy similarity metric between student agent i's policy π_i and teacher policy π^j_* as evaluation for student's skill on sub-task T^j. Then, we formulate the sub-task assignment as a bandit problem to address the trade-off between exploiting the best-known assignment and exploring less-tested assignments that could potentially better leverage the students' skills and lead to improved team performance. Note that in Phase 2, the student agents will be advised by the teacher based on their assigned sub-tasks as identified in Phase 1.

1) Policy similarity metric: We first introduce the cross-entropy between the student policy and the sub-task policy to define policy similarity. The (negative) cross-entropy from student policy π_i to a sub-task policy π_*^j at state s is defined as

$$\mathrm{CE}(\pi_i, \pi_*^j, s) = \mathbb{E}_{a\sim\pi_i(\cdot|s)}[\log(\pi_*^j(a \mid s))], \tag{3.1}$$

representing an estimated (negative) distribution discrepancy from agent i's policy to the optimal policy of sub-task T^j at state s. We define the policy similarity over a state distribution $d(s)$ as the expected cross-entropy measured in $d(s)$ based on the above strategy:

$$h_{d(s)}(\pi_i, \pi_*^j) = \mathbb{E}_{s\sim d(s)}[\mathrm{CE}(\pi_i, \pi_*^j, s)], \tag{3.2}$$

Intuitively, it would be better to adapt π_i to π_*^j than to π_*^k in $d(s)$ if π_i is more similar to policy of sub-task T^j than that of sub-task k, i.e., $h_{d(s)}(\pi_i, \pi_*^j) > h_{d(s)}(\pi_i, \pi_*^k)$.

2) Guided sampling for policy similarity estimation: The choice of $d(s)$ has a significant influence on the metric. It is essential to cover the states where sub-tasks operate in to get sufficient information for skill evaluation. There is a risk of overestimation or underestimation if $d(s)$ only covers the states where the agent is bad or good at certain skills. Therefore, we propose a guided sampling strategy in which the student agent is guided by the policies in Π_* to sample adequately in the state space. The policy similarity metric we use can be written as

$$h_{\tilde{\pi}_i}(\pi_i, \pi_*^j) = \mathbb{E}_{s\sim\tilde{\pi}_i}[\mathrm{CE}(\pi_i, \pi_*^j, s)], \tag{3.3}$$

where $\tilde{\pi}_i$ is the advised sampling policy similar to that introduced in Sec. 2.2, which will be described in detail in Sec. 3.3. Note a slight notation abuse for simplicity here, the state distribution sampled by policy $\tilde{\pi}_i$ is directly represented by $\tilde{\pi}_i$ .We verify this idea empirically in Sec. 4.3. This policy similarity metric is further used to define the reward function for the MAB.

3) MAB-based sub-task assignment: Now we formulate the sub-task assignment as an MAB problem iterating for M games, where M represents the total game count of phase 1. The problem is considered over a bandit space defined as

$$B = \{\{b_1, \cdots, b_n\} \mid \ b_i \in \mathcal{T},\ b_j \in \mathcal{T},\ b_i \neq b_j,\ \ \forall\, i, j \in N\} \tag{3.4}$$

where b_i represents the assigned sub-task for agent i. The bandit space is constrained such that each agent will be assigned with unique sub-tasks in $\mathcal{T}$. For each game, the teacher agent selects an assignment $b \in B$ and uses the guided sampling strategy to evaluate students' skills. The advising policy for each student corresponds to the optimal policy of the student's assigned sub-task, while the teacher agent directly executes its own sub-task. This way, the teacher agent is able to estimate students' policy similarity to different sub-tasks sufficiently over the state space and receives a reward based on the estimated policy similarity.

The reward is formulated as

$$r(b) = \sum_{i \in N_{-1}} h_{\tilde{\pi}_i}(\pi_i, \pi_*^{b_i}) \tag{3.5}$$

which considers the skill evaluation of all students for the assignment b. In this paper, we adopt the Upper Confidence Bound (UCB) [26] algorithm as our assignment strategy. For each game $m \in \{1, \cdots, M\}$, the optimal assignment b^* is computed as

$$b^* = \arg\max_b \begin{cases} r(b) + \mu\sqrt{\dfrac{2\log m}{m_b}}, & m \leq M \\ r(b), & m > M \end{cases} \tag{3.6}$$

where m_b denotes the frequency bandit b is sampled before game m, μ is a tuned hyper-parameter that controls the weight of the confidence bound term [26] and balances exploration and exploitation. With higher μ, the exploration component is magnified, and the strategy leans to select assignments that are less tried. After M games, the bandit reward is no longer updated, and the system follows the best-know assignment in Phase 2.

4) Implementation: We provide the pseudo-code for algorithm implementation of Phase 1 in Alg. 1. For simplicity, teacher agent is denoted as agent 1 and student agents are agent 2 to N. At the beginning of each game, the teacher agent assign sub-task b^* to all agents according to the Bandit algorithm. For each step, students first send their intentional actions to the teacher for policy-similarity estimation, and receive action advice from the teacher agent based on the *Action Advise* algorithm which will be introduced in Sec. 3.3. Then agents interact with the environment

synchronously, where the teacher agent executes its own sub-task b_1^* with optimal action $a_*^{b_1}$. The transition is collected into $\mathcal{D}_b$ for bandit reward update, and each student agent's train batch $\mathcal{D}_i$ for its own policy update. After each game, the teacher agent updates the bandit reward with Eq. 3.1, 3.3, 3.5.

Algorithm 1 Phase 1: Sub-Task Assignment

In first M games
//Sample
for each game **do**
  Initialize each student i trajectory batches for RL $\mathcal{D}_i = \emptyset$, Agent 1's batch for bandit reward update $\mathcal{D}_b = \emptyset$
  Teacher samples tasks assignment b^* based on Equation 3.6
  for each environment step τ with state s_τ **do**
    Each agent i sends intentional action $a_i \sim \pi_i(\cdot \mid s_\tau)$ to the teacher agent
    student i's sampling action $\tilde{a}_i = \textit{Action Advise}(s, \tau, b_i^*, a_i)$
    Agents interact with the environment using joint action $(a_*^{b_1}, \tilde{a_2}, \cdots, \tilde{a_N})$
    $\mathcal{D}_b \leftarrow \mathcal{D}_b \bigcup (a_i, s_\tau)$, $\mathcal{D}_i \leftarrow \mathcal{D}_i \bigcup (s_\tau, a_i, s_{\tau+1}, R_\tau)$
  end for
end for
//Update
Update $r(b^*)$ using $\mathcal{D}_b$ with Eq. 3.1, 3.3, 3.5
Each agent i updates π_i, using trajectories in $\mathcal{D}_i$

3.3 Phase 2: Advising

With our MAB-based sub-task assignment using policy similarity-based rewards after M games in Phase 1, the student agents are assigned sub-tasks that fully leverage their existing skills. In this phase, they continue receiving advice to refine these skills and further improve team learning performance. We first introduce the universal formulation of action advising strategy in both phases, while in Phase 1 it serves as the guided sampling for policy similarity estimation in Eq. 3.3. Then we describe Phase 2 in detail.

1) Action advising strategy: As discussed in Sec. 2.2, student i's sampling policy with action advice can be expressed by incorporating teaching policy with an

advice-following indicator I as

$$\tilde{\pi}_i(\cdot \mid s) = I \cdot \pi_*^{b_i}(\cdot \mid s) + (1 - I) \cdot \pi_i(\cdot \mid s) \tag{3.7}$$

where b_i is the assigned sub-task to student i in a specific game. As a result of exploration in the sub-task assignment, $\tilde{\pi}_i$ varies in Phase 1 according to Eq. 3.6, while consistent through Phase 2. We adopted the idea of safe-guarded exploration proposed in [49] and advice-following indicator I is sampled as

$$I \sim p(\pi_i, \pi_*^j, s, \tau) = \begin{cases} 0, & \text{if} \quad \mathrm{CE}(\pi_i, \pi_*^j, s) > \eta \\ \alpha_0(1 - \tau/\tau_{\max}), & \text{else} \end{cases} \tag{3.8}$$

where $\eta < 0$ is a tuned threshold, above which the student's policy is considered similar to the teacher policy π_*^j at s, therefore regarded proficient at sub-task T^j; hence advice not issued. Similar to what described in Phase 1, the cross-entropy here is computed using student's intentional action. The advice probability is annealed throughout the learning process to enable sufficient exploration of the student agent to learn a robust policy [43]. $\alpha_0 < 1$ is the initial advising probability, which decays to 0 linearly in $\tau_{\max}$ environment steps. For every environment step, the teacher agent samples I based on the current advising probability to decide whether to issue advice to each student. The students follow the teacher's advising policy if advice is given.

The implementation of action advising algorithm is given in Alg. 2.

Algorithm 2 *Action Advise*

Require: Intentional action a_i, sub-task b_i, state s, current time step τ
  Teacher agent compute cross-entropy $\mathrm{CE}(\pi_i, \pi_*^{b_i}, s)$ according to Eq. 3.1
  Sample advice-issuing indicator I according to Eq. 3.8
  if $I \neq 0$ **then**
    Teacher agent samples $\tilde{a}_i \sim \pi_*^{b_i}(\cdot \mid s)$, send action advice to agent i
  else
    Agent i samples $\tilde{a}_i \sim \pi_i(\cdot \mid s)$
  end if

2) Phase 2: With action advice defined above, the student agents are able to efficiently explore well-rewarded samples for a consistent sub-task in Phase 2. And the samples are used to boost their policy training to maximize the team objective. In

this work specifically, IPPO described in Sec. 2.2 is used though the action advising method is agnostic to students' learning algorithms. Similar to Phase 1, the teacher agent still follows a policy from its library corresponding to its assigned sub-task and is not being trained. The algorithm implementation of Phase 2 is shown in Alg. 3.

Algorithm 3 Phase 2: Advising

In games after the first M games
Consistent sub-task assignment b^* based on Equation 3.6
//Sample
for each game **do**
  Initialize each student i trajectory batches for RL $\mathcal{D}_i = \emptyset$
  for each environment step τ with state s_τ **do**
    Each agent i sends intentional action $a_i \sim \pi_i(\cdot \mid s_\tau)$ to the teacher agent
    student i's sampling action $\tilde{a}_i = \textit{Action Advise}(s, \tau, b_i^*, a_i)$
    Agents interact with the environment using joint action $(a_*^{b_1}, \tilde{a_2}, \cdots, \tilde{a_N})$
    $\mathcal{D}_i \leftarrow \mathcal{D}_i \bigcup (s_\tau, a_i, s_{\tau+1}, R_\tau)$
  end for
end for
//Update
Each agent i updates π_i with IPPO, using trajectories in $\mathcal{D}_i$

Chapter 4

Experiments

In this chapter, we empirically evaluate the effectiveness of our experience-based action advising approach in Grid World with 2-agent and 3-agent scenarios and in Overcooked-AI [7] with a 3-agent scenario. First we give the experiment setups for three testing scenarios in the Grid World and Overcooked-AI in Sec. 4.1, including the details of our teacher agents and student agents. Then we provide experiment results starting from Sec. 4.2. By comparing with three baselines, we answer the following questions: 1) How does our method compare to action advising without considering an agent's prior knowledge? 2) How does our method compare to no advising? 3) How does our method comparing with adapting teacher agent's policy to accommodate for students' incompetency? We also conducted ablation studies to evaluate the importance of each components in our proposed method.

4.1 Experiment Setup

Grid World: A discrete urban search and rescue (USAR) task [18, 19] is adopted for the Grid World scenarios. In the 2-agent scenario, there are three pieces of rubble (yellow dots) and two victims (red crosses). The agents (triangles) should heal both victims and clear all rubble to receive a team-level reward. We pre-decomposed the task into role-based sub-tasks: *"remove the rubble"* and *"heal the victim."* This task requires efficient collaboration between the agents, as one victim is blocked by two pieces of rubble. In the 3-agent scenario, the other victim is locked behind a door

(green block) that can be unlocked with a key. Therefore, the third sub-task, *"open the door,"* is required to be accomplished by the third agent.

Overcooked-AI: Three agents work in a kitchen to cook and deliver soups with three recipes: "onion," "tomato," and "onion and tomato." The soups should be delivered consecutively, following the recipe order. Agents receive a team-level reward every time soups of all three recipes are correctly delivered. Three sub-tasks are decomposed for "onion chef," "tomato chef", and "service agent". To cook and deliver the "onion soup", the "onion chef" should pick up an onion from the onion dispenser and place it in the pot (dark grey) to start cooking. After cooking finishes in five steps, the "service agent" should grab a plate from the dish dispenser, pick up the cooked soup from the pot, and deliver it to the service counter (light grey). The same process is needed for the "tomato soup." For the "onion and tomato soup," both the "onion chef" and "tomato chef" should bring their respective ingredients to the pot. For each of these three environments, the teacher agent is pre-trained to acquire

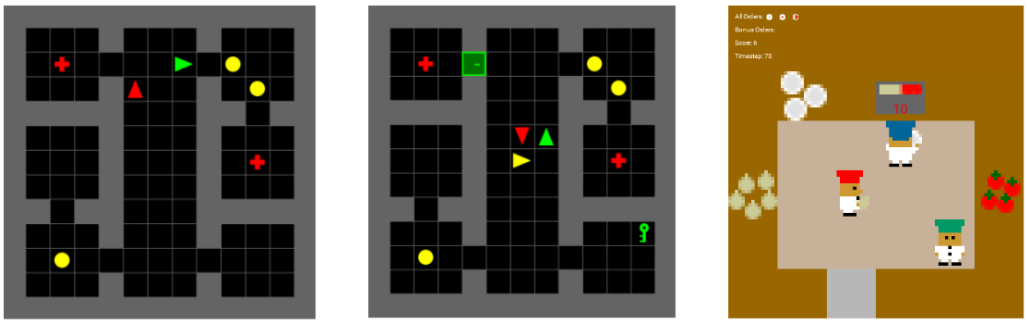

Figure 4.1: Layout for 2-agent Grid World (left) and 3-agent Grid World (middle). Yellow balls are rubble, and red plusses are victims. Overcooked-AI (right) with onions, tomatoes, plates, and a stove.

a policy library containing optimal policies for each of the decomposed sub-tasks with individual rewards. To train on the individual sub-task, optimal agents for other sub-tasks are employed to accomplish necessary pre-conditions. For example, in the 3-agent Overcooked-AI scenario, optimal "tomato chef" and "onion chef" accompany the teacher when it is training for the "service agent" sub-task to make soups for the teacher agent. The individual rewards are defined based on goal-reaching, which will only be issued when the teacher agent's sub-task is accomplished. For instance, in the 3-agent Grid World scenario, the teacher will receive rewards after the door is

opened when training for the *"open the door"* sub-task.

Multiple student agents with varying initial skills are tested in each scenario. For convenience, a student agent is named based on an abbreviation of sub-tasks and its initial success rate on the sub-tasks. For instance, in the USAR experiments, 'R50' indicates that the agent is partially trained for "remove the rubble", and is able to accomplish the sub-task with a success rate of 50% when its teammates follow optimal policies for other sub-tasks, but is not able to accomplish 'heal the victim'. Similarly, 'O50+H5' refers to a student agent in the 3-agent task, representing the agent has skills for 'open the door', abbreviated as 'O', with 50% success rate. And the agent has a trivial skill with 5% success rate for "heal the victim". In 3 agent teams where there are two students, name of each student is separated with "&". Details about the sub-tasks and abbreviations are in Table 4.1. Without losing generality, the initial skills are trained from individual rewards similar to that of the teacher agent, but the training process is cut off before convergence to the optimal skills to have sub-optimal students.

Table 4.1: sub-tasks and abbreviations for each scenario

	sub-task 1	sub-task 2	sub-task 3
Grid World 2 agent	Remove the rubble (R)	Heal the victim (H)	
Grid World 3 agent	Remove the rubble (R)	Heal the victim (H)	Open the door (O)
Overcooked 3 agent	Onion chief (O)	Tomato chief (T)	Service agent (S)

In Sec. 4.2, we compare our approach with the following baselines, and 5 trials are run for each experiment.

- **Advising with an arbitrary assignment**, where the teacher agent samples an arbitrary assignment $b \in B$ at the beginning of the training process, and hereafter teaches each student i following the Phase 2 algorithm. The student's prior experience is not considered.
- **No advising**, where the teacher agent only assigns a sub-task for itself based on its identification of the teammate's initial skills with the bandit formulation but does not teach the student agent. This resembles the idea in ad-hoc teamwork [5], and hence called as **naive ad hoc teamwork baseline**.

- **Multi-agent reinforcement learning (MARL)** in three-agent experiments, where the teacher agent and student agents jointly learn with IPPO algorithm under an arbitrary assignment $b \in B$ sampled at the beginning of the training process. Specifically, the teacher adapts its optimal sub-task policy while students continue to learn with their existing skills. No advice is issued.

In Sec. 4.3, ablation studies are conducted to examine the rationale our sub-task assignment algorithm based on 1) UCB-based bandit formulation, 2) guided sampling for policy similarity estimation.

4.2 Main Results

4.2.1 Two-Agent Grid World

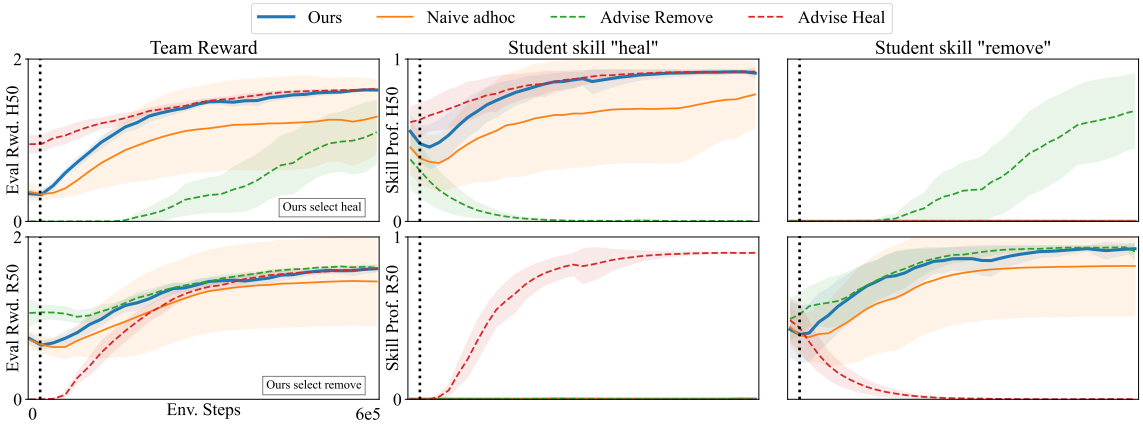

Figure 4.2: Evaluation for Grid World 2 agent setting. The black dotted line represents the end of Phase 1 of the proposed method. The first column illustrate team performance for student agent "H50" and "R50". The rest of two columns shows how each skill of each student evolves along the training process.

Student agents that initially have limited prior knowledge of *"remove the rubble"* and *"heal the victim"* are tested. In Fig 4.2, black dotted lines represent the end of the sub-task assignment phase. The dashed curves represent baseline **advising with an arbitrary assignment**. The first column illustrate team performance for each student agent and the rest of two columns shows how each skill of each student evolves along the training process. In the case of "H50" (Fig 4.2 row 1) where a student

is with knowledge of *'heal the victim"*, it would be optimal to teach the student to heal the victim. Therefore the upper bound of the performance is ideally assigning and advising with the heal task, corresponding to the red dashed line in Fig 4.2 (row 1). Conversely, asking this student to perform the task of removing rubble is the lower bound (green dashed curves). In team reward evaluation (column 1), the curve of our method (blue curve) converges to the upper bound gradually, demonstrating that our targeted action advising based on experience can achieve fast convergence with students' initial expertise leveraged. Meanwhile, our method outperforms the **naive ad hoc** baseline (orange curve) not significantly, as the sub-tasks are relatively easy in the 2-agent scenario. Both our method and the naive ad hoc baseline which requires sub-task assignment have a smaller initial team reward due to exploration with sub-optimal sub-task assignments. From the evaluation of student skill *"heal the victim"*, we can tell that this sub-task assignment phase also causes slight decrease of the student agent's proficiency on its skill, which explains the recover of team reward after phase 1. And advising the student with the wrong sub-task will force it de-learn its existing expertise (green curve in column 2, row 1), and slowly learns the new skill of removing the rubble (green curve in column 3, row 1). The result for student "R50" shown in Fig.4.2 (row 2) demonstrates similar trend with the switched performance upper and lower bounds, as the student possesses initial knowledge on *"remove the rubble"*.

The result shows our method's effectiveness in a simple 2-agent setting. However, the bandit space B is trivial in this scenario with only two possible bandit arms, making it necessary to test in more complex multi-agent games where more agents and sub-tasks are presented, increasing the difficulty of both sub-task assignment and student learning.

4.2.2 Three-Agent Experiments

Now we test our method in more complex three agent settings. For these experiments, all six arbitrary assignments in B are tested to compare with our method, with each assignment corresponding to a numerical notation. The relationship between numerical notation and agent sub-tasks is shown in Tab. 4.2.

Table 4.2: sub-task notation and corresponding assignments in 3-agent tasks

	Grid World			Ovorcooked		
	Teacher	Student 1	Student 2	Teacher	Student 1	Student 2
1	Remove	Heal	Open	Onion	Tomato	Serve
2	Remove	Open	Heal	Onion	Serve	Tomato
3	Heal	Remove	Open	Tomato	Onion	Serve
4	Heal	Open	Remove	Tomato	Serve	Onion
5	Open	Remove	Heal	Serve	Onion	Tomato
6	Open	Heal	Remove	Serve	Tomato	Onion

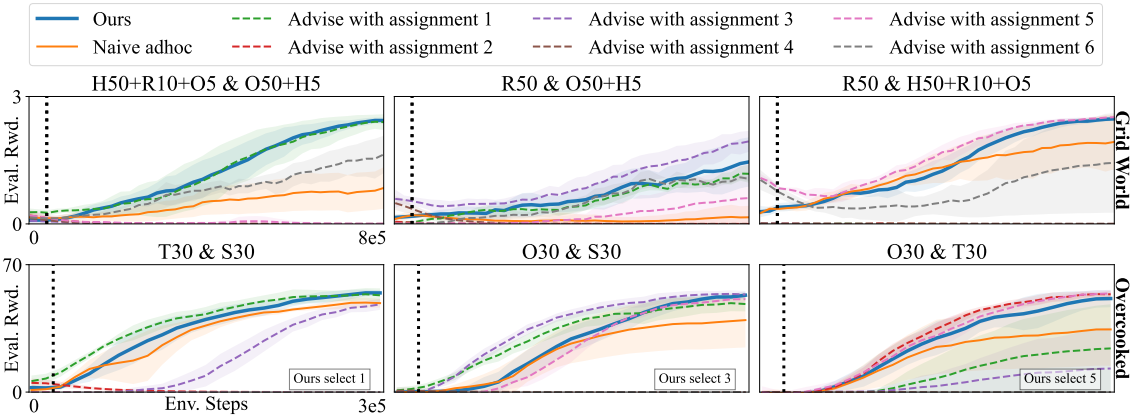

Figure 4.3: Team performance evaluation for 3 agent scenarios. Semantic meaning of each assignment in Table 4.2. The black dotted line represents the end of Phase 1 of the proposed method.

We provide team reward evaluation results in Figure 4.3. Similar to the result of 2 agents, the student agents are assigned and taught with the most efficient sub-tasks as shown in Figure 4.3. In 3-agent Grid World (Figure 4.3 row 1), performance with optimal assignment for the "R50 & O50+H5" agent (column 2) increases slowly, implying an inherent difficulty in credit assignment for students to learn "remove the rubble" and "open the door" together, which serve as pre-conditions for the agents to accomplish the task and receive team reward [46]. Our method failed to converge to the same performance as that of optimal assignment, emphasizing a trade-off of our method, where appropriate sub-tasks are identified and assigned at the cost of exploration with inappropriate sub-tasks, which leads to students' skill degradation in

the sub-task assignment phase. Due to the exploration complexity of multi-agent RL, our method exceeds the performance of the naive ad hoc teamwork baselines. The result reveals that our method can identify student's skills in a complex multi-agent setting, and effectively teach students with sub-tasks that are relatively efficient for them to learn, subsequently improving the overall team learning efficiency.

In 3-agent Overcooked experiments (Figure 4.3 row 2), there are two assignments with similar optimality regarding team learning efficiency for student groups "O30 & S30" (middle figure) and students "O30 & T30". In these cases, our method can constantly assign the sub-task on which the student has a larger policy similarity. For example, in the "O30 & S30" group, the "O30" agent is assigned and educated as "onion chief" and "S30" as "service". The other similarly optimal assignment, educating the "O30" agent with the "tomato chief" sub-task in this case, is not selected due to the lower similarity of the "O30" agent's "onion chief" skill for the "tomato chief" sub-task. This implies a disproportional relationship between agents' policy similarity to sub-tasks and potential learning efficiency on the sub-tasks, where smaller policy similarity is not guaranteed to yield worse learning efficiency. This raises a potential future investigation direction on better quantifying the relationship between policy similarity and learning efficiency.

The experiments above emphasizes our method's ability to have the teacher agent helping students' learning to improve the team performance. How does it compare with adapting the teacher agent as complement for students' incapability to help the team?

We compare our method with MARL baselines where teacher agent also learns from its sub-task policy when students refine their existing skills. Results are shown in Figure 4.4. Since agents learn with team reward in our fully cooperative games, sub-task assignment 1 in Table 4.2 is essentially the same as 2 since the teacher agent takes the *"remove the rubble"* or "onion chef" sub-task and adapts from the corresponding optimal policy, while students are learning on their own. Similarly, sub-task assignment 3 is the same as 4, 5 is the same as 6. Therefore, we compare our method with MARL baselines where teacher and students jointly learn with sub-task assignment 1, 3, and 5. In the Grid World scenario (Figure 4.4 row 1), MARL fails to learn effective policies in any of the cases. The result indicates the difficulty of credit assignment in this task, and the necessity of knowledge transfer

tailored to student agents' initial skills for role-based collaboration. For instance, when student agents possess skills for *"remove the rubble"* and *"heal the victim"* initially and teacher adapts its policy for *"open the door"* (column 3 red dashed curve), the team performance constantly drops as students are confused with which role to learn in the fully cooperative game, and the teacher's adaptation cannot accommodate all necessary skills for the task. In the Overcooked Scenario (Figure 4.4 row 2), the teacher's adaptation without tailored advising reaches performance similar to our method under optimal sub-task assignments. In this scenario, students can continue to refine their skills without the help of advice, underlying an relatively easy exploration. And the teacher learns to compensate for the student's incompetence at the beginning, reaching even faster increase of team performance than ours. The result implies the possibility of further improving the performance by integrating our method with teacher's policy adaptation. The result also demonstrate the importance of skill identification, as in both scenarios, students fail to learn when the sub-task corresponds to their initial skills is occupied by the teacher.

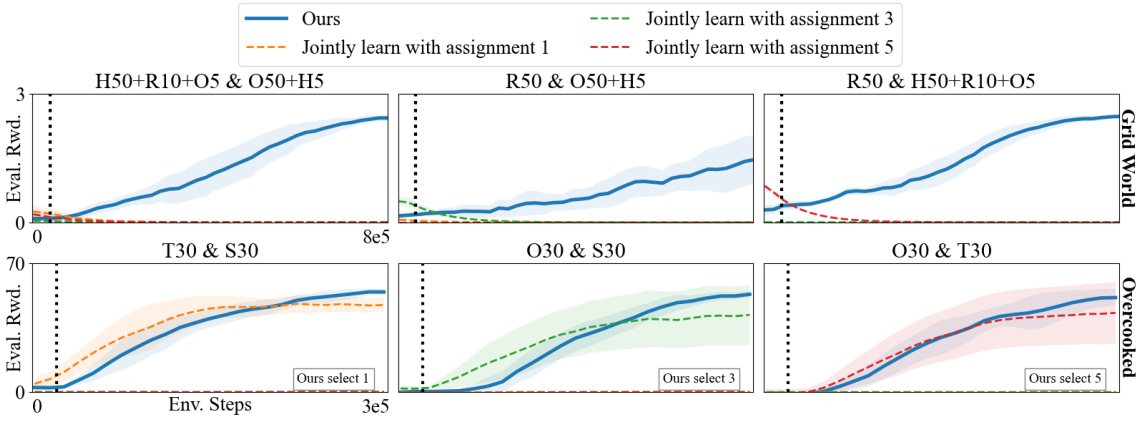

Figure 4.4: Team performance evaluation for 3 agent scenarios comparing with MARL. Semantic meaning of each assignment in Table 4.2. The black dotted line represents the end of Phase 1 of the proposed method.

4.2.3 Takeaway

The main results demonstrate that our method is able to significantly improve the team's performance by tailoring action advice to students based on their initial skills and task requirements. Compared with **advising with an arbitrary assignment**

and **naive ad hoc** baselines, we showcase the importance of assigning appropriate roles to students based on skill identification and the effectiveness of action advice on further expediting the team's learning. Specifically, the skill evaluation curves in Figure 4.2 show the advantage of our method by avoiding forcing students to forget existing skills. By comparing with **MARL** baseline in three-agent scenarios, we illustrate that while adapting teacher's policy can help the team's performance initially by constantly acquiring rewards, students cannot learn effectively to help improve the final team performance. This indicates the value of role-based collaboration in these fully cooperative multiagent games as the teacher cannot takeover all sub-task skills on itself, and further underlines the necessity of knowledge transfer based on students' experience.

4.3 Ablation

In this section we study how components in Phase 1 influence the sub-task assignment and students' learning. Specifically, we investigate the influence of bandit formulation and guided sampling by address the following two questions:

- **MAB formulation:** How does the UCB-based bandit formulation contribute to the improved efficiency?
- **Guided sampling:** How does the guided sampling contribute to students' expertise identification?

All ablation studies take place in the Grid World and Overcooked three-agent scenarios.

4.3.1 MAB Formulation

We employ an MAB formulation with the UCB algorithm to effectively balance exploration and exploitation for our task assignment. While excessive exploration might be helpful to sufficiently evaluate students' prior skills, leveraging the current best skill is also important. To understand the trade-off and demonstrate the effectiveness of the bandit formulation, we conducted an ablation study comparing our UCB-based method with a uniform exploration strategy. In the uniform strategy, task assignments are sampled uniformly from the bandit space B in the first M games to ensure a broad exploration of possible assignments. After these initial M games,

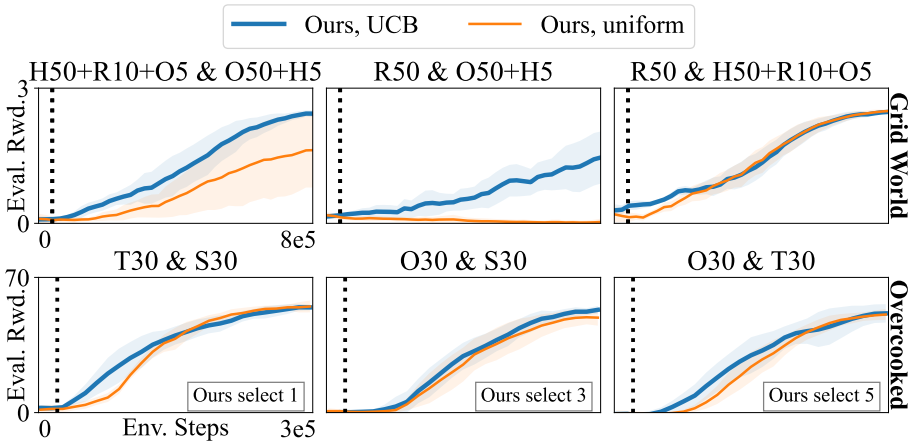

Figure 4.5: 3-agent scenarios bandit formulation ablation results—Phase 1 of the proposed method.The black dotted line represents the end of Phase 1 of the proposed method.

the assignment with the highest estimated reward is consistently selected:

$$b \begin{cases} \sim \text{Uniform}(B), & m \leq M \\ = \arg\max r(b), & m > M \end{cases} \tag{4.1}$$

The result in 3-agent Grid World scenario is shown in Figure 4.5. Our UCB-based bandit formulation outperforms the uniform exploration strategy in both sample efficiency and convergence rewards. While the uniform approach treats all assignments equally, the UCB algorithm dynamically prioritizes assignments based on estimated rewards and uncertainties. This adaptive process allows the UCB method to gradually filter out suboptimal task assignments. The improved convergence of our method further underscores the importance of efficient sub-task assignment for our problem in preventing students from forgetting their prior expertise [48], which can occur from prolonged engagement with unfamiliar sub-tasks. The uniform exploration leads to a similar performance to ours for Overcooked experiments, as the estimated $r(b)$ has a smaller difference among assignments, leading to encouraged exploration. Therefore, our UCB-based algorithm is not able to exploit the best assignment significantly more than the uniform exploration, as shown in Figure 4.6. This finding underlies the importance of MAB formulation with exploration strategies that can

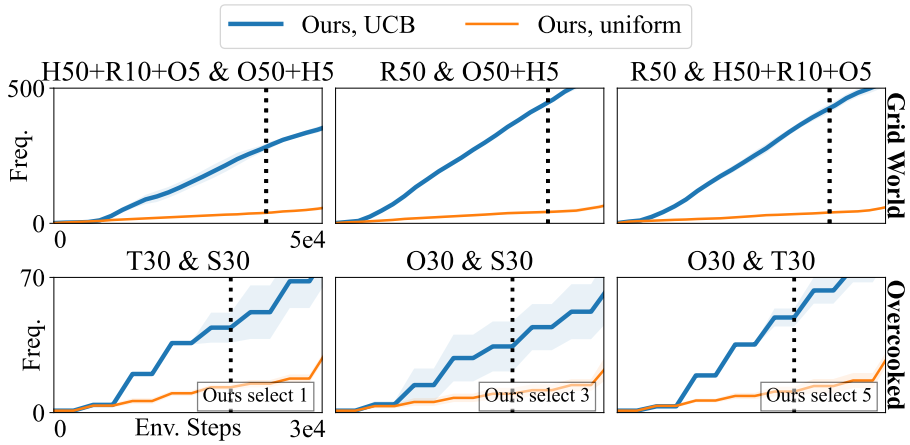

Figure 4.6: Frequency of optimal sub-task assigned in 3-agent scenarios for uniform exploration ablation study. The black dotted line represents the end of Phase 1 of the proposed method.

balance exploration on less tried sub-task assignments and exploitation of already well-estimated best assignment.

4.3.2 Guided Sampling

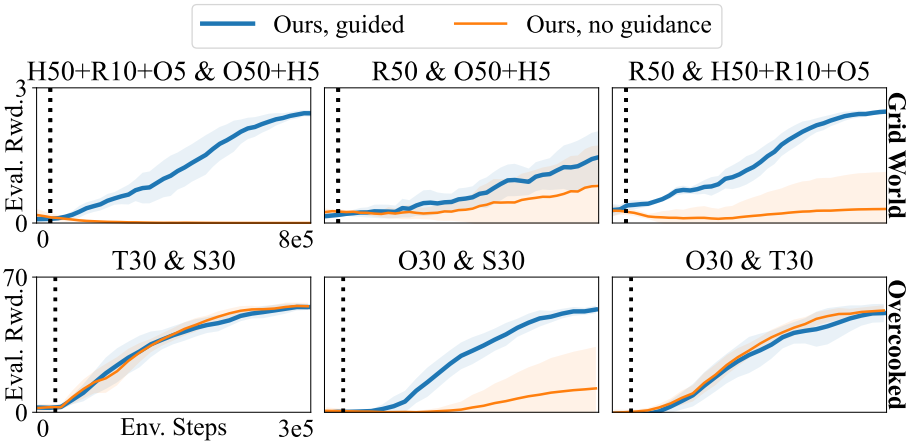

Figure 4.7: 3-agent scenarios guided sampling ablation results—Phase 2 of the proposed method. The black dotted line represents the end of Phase 1 of the proposed method.

Intuitively, guided sampling can help student agents sample more thoroughly over the state space and provide a better evaluation of students' sub-task skills based on the policy similarity metric. To verify this intuition, we compare our algorithm with a variant where policy similarity is estimated without guided sampling, where the policy similarity between agent i and sub-task T^j is estimated as $h_{\pi_i}(\pi_i, \pi_*^j)$. That is to say, we manually set advising probability to 0 for any agent i in the sub-task assignment phase, and restore the computed $p(\pi_i, \pi_*^j, s, \tau)$ at the start of advising phase. Without guidance, the algorithm failed to distinguish student agents' skills except in the first and third case of 3 agent Overcooked scenario. For example, the "H50+R10+O5" agent in Grid World is constantly educated with sub-task "remove the rubble", leading to the suboptimal team learning performance due to this inappropriate sub-task assignment. The result demonstrates the necessity of guided sampling for distinguishing students' pre-existing skills.

Chapter 5

Conclusions

We introduce an experience-based action advising framework for knowledge transfer in multi-agent teams with varying experience levels to enhance teaming efficiency. Our algorithm leverages teacher policies to assess students' existing expertise and assigns appropriate sub-tasks using a multi-armed bandit approach that balances exploration and exploitation. Subsequently, students are educated on these sub-tasks in alignment with their skills and task role demands to improve the team's learning performance. Empirical results show that our method converges more efficiently than methods that ignore students' prior knowledge when advising and outperforms teams without knowledge transfer. In addition, our experience-based action advising is more effective at promoting learning of collaboration, compared to methods in which the teacher agent simply adapts its behavior to accommodate students' incompetency. Ablation studies further validate the superiority of our MAB formulation with UCB strategy and guided sampling for policy similarity estimation. These findings together highlight the effectiveness of our approach in increasing the learning efficiency of mixed-experience multi-agent teams.

Despite these contributions, our method has certain limitations that open multiple avenues for future investigation:

Teacher policy library learning: Our method relies on a strong assumption of decomposed sub-tasks with individual rewards in Sec. 3.1 to train the teacher agent. This is not always feasible especially for tasks where sub-tasks interleave for effective collaboration. Consequently, an interesting direction is to develop an algorithm that

directly learns role-based collaboration policy library in a fully cooperative game. Existing techniques in diversity MARL [28] and role discovery MARL [45] could inform such a solution. Additionally, our current method is constrained to teams where the number of agents matches the policy library size, due to exact sub-task decomposition. Defining teacher policy libraries instead as collections of skills and skill selectors [27, 50] might circumvent this issue by separately considering and transferring individual skills and collaboration knowledge to the students.

Sub-task assignment complexity and skill evaluation: The current MAB-based sub-task assignment has combinatorial complexity with respect to the number of sub-tasks, which can be inefficient at scale. Pre-training could help shift part of this sampling burden away from online estimation. The ad-hoc teamwork literature (e.g., [8, 38]) offers potential pathways, as the teacher agent could learn from multiple student teams before collaborating with the target team, thus developing more effective "teaching experience." This could also pave the way for stronger skill evaluation metrics. As noted in Section 4.2, the relationship between a student's learning efficiency for a sub-task and its policy similarity can be disproportionate. Further investigation is needed to quantify and leverage these relationships more accurately for knowledge transfer.

knowledge transfer in ad hoc teamwork: Our framework addresses knowledge transfer for ad hoc teacher-student teams under the assumption of a shared communication protocol for actions. An important question for future work is how to enable knowledge transfer when no such communication is available. Approaches based on behavior influence [47] and social influence [22] might be promising, but balancing the teacher's behavior between influencing students and achieving team-wide rewards remains both critical and challenging.

Bibliography

[1] Stefano V Albrecht and Subramanian Ramamoorthy. A game-theoretic model and best-response learning method for ad hoc coordination in multiagent systems. *arXiv preprint arXiv:1506.01170*, 2015. 2.1

[2] Yanwen Ba, Xuan Liu, Xinning Chen, Hao Wang, Yang Xu, Kenli Li, and Shigeng Zhang. Cautiously-optimistic knowledge sharing for cooperative multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17299–17307, 2024. 2.1

[3] Goonmeet Bajaj, Sean Current, Daniel Schmidt, Bortik Bandyopadhyay, Christopher W Myers, and Srinivasan Parthasarathy. Knowledge gaps: A challenge for agent-based automatic task completion. *Topics in Cognitive Science*, 14(4): 780–799, 2022. 1

[4] Samuel Barrett and Peter Stone. Cooperating with unknown teammates in complex domains: A robot soccer case study of ad hoc teamwork. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015. 2.1

[5] Samuel Barrett, Avi Rosenfeld, Sarit Kraus, and Peter Stone. Making friends on the fly: Cooperating with new teammates. *Artificial Intelligence*, 242:132–171, 2017. 1, 2.1, 4.1

[6] Joseph Campbell, Yue Guo, Fiona Xie, Simon Stepputtis, and Katia Sycara. Introspective action advising for interpretable transfer learning. In *Conference on Lifelong Learning Agents*, pages 1072–1090. PMLR, 2023. 2.2

[7] Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32, 2019. 1, 4

[8] Shuo Chen, Ewa Andrejczuk, Zhiguang Cao, and Jie Zhang. Aateam: Achieving the ad hoc teamwork by employing the attention mechanism. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7095–7102, 2020. 2.1, 5

[9] Filippos Christianos, Georgios Papoudakis, Muhammad A Rahman, and Stefano V Albrecht. Scaling multi-agent reinforcement learning with selective

parameter sharing. In *International Conference on Machine Learning*, pages 1989–1998. PMLR, 2021. 2.1

[10] Felipe Leno Da Silva and Anna Helena Reali Costa. A survey on transfer learning for multiagent reinforcement learning systems. *Journal of Artificial Intelligence Research*, 64:645–703, 2019. 1

[11] Felipe Leno Da Silva, Ruben Glatt, and Anna Helena Reali Costa. Simultaneously learning and advising in multiagent reinforcement learning. In *Proceedings of the 16th conference on autonomous agents and multiagent systems*, pages 1100–1108, 2017. 2.1

[12] Felipe Leno Da Silva, Garrett Warnell, Anna Helena Reali Costa, and Peter Stone. Agents teaching agents: a survey on inter-agent transfer learning. *Autonomous Agents and Multi-Agent Systems*, 34:1–17, 2020. 2.1, 2.2

[13] Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020. 2.2

[14] Anestis Fachantidis, Matthew E Taylor, and Ioannis Vlahavas. Learning to teach reinforcement learning agents. *Machine Learning and Knowledge Extraction*, 1 (1):21–42, 2017. 2.1

[15] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 2.2

[16] Elliot Fosong, Arrasy Rahman, Ignacio Carlucho, and Stefano V Albrecht. Learning complex teamwork tasks using a given sub-task decomposition. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 598–606, 2024. 2.1

[17] Pengjie Gu, Mengchen Zhao, Jianye Hao, and Bo An. Online ad hoc teamwork under partial observability. In *International conference on learning representations*, 2021. 2.1

[18] Yue Guo, Rohit Jena, Dana Hughes, Michael Lewis, and Katia Sycara. Transfer learning for human navigation and triage strategies prediction in a simulated urban search and rescue task. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, pages 784–791. IEEE, 2021. 4.1

[19] Yue Guo, Joseph Campbell, Simon Stepputtis, Ruiyu Li, Dana Hughes, Fei Fang, and Katia Sycara. Explainable action advising for multi-agent reinforcement learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5515–5521. IEEE, 2023. 2.1, 4.1

[20] Ercüment Ilhan, Jeremy Gow, and Diego Perez-Liebana. Teaching on a budget in multi-agent deep reinforcement learning. In *2019 IEEE Conference on Games (CoG)*, pages 1–8. IEEE, 2019. 1, 2.1

[21] Vidhi Jain, Rohit Jena, Huao Li, Tejus Gupta, Dana Hughes, Michael Lewis, and Katia Sycara. Predicting human strategies in simulated search and rescue task. *arXiv preprint arXiv:2011.07656*, 2020. 1

[22] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*, pages 3040–3049. PMLR, 2019. 5

[23] Dong-Ki Kim, Miao Liu, Shayegan Omidshafiei, Sebastian Lopez-Cot, Matthew Riemer, Golnaz Habibi, Gerald Tesauro, Sami Mourad, Murray Campbell, and Jonathan P How. Learning hierarchical teaching policies for cooperative agents. *arXiv preprint arXiv:1903.03216*, 2019. 2.1

[24] Woojun Kim and Youngchul Sung. An adaptive entropy-regularization framework for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 16829–16852. PMLR, 2023. 2.2

[25] Woojun Kim, Whiyoung Jung, Myungsik Cho, and Youngchul Sung. A variational approach to mutual information-based coordination for multi-agent reinforcement learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 40–48, 2023. 2.2

[26] Volodymyr Kuleshov and Doina Precup. Algorithms for multi-armed bandit problems. *arXiv preprint arXiv:1402.6028*, 2014. 2.2, 3.2, 3.2

[27] Youngwoon Lee, Jingyun Yang, and Joseph J Lim. Learning to coordinate manipulation skills via skill behavior diversification. In *International conference on learning representations*, 2019. 5

[28] Chenghao Li, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:3991–4002, 2021. 1, 5

[29] Yongyuan Liang and Bangwei Li. Parallel knowledge transfer in multi-agent reinforcement learning. *arXiv preprint arXiv:2003.13085*, 2020. 2.1

[30] Laetitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *The Knowledge Engineering Review*, 27(1):1–31, 2012. 2.2

[31] Francisco S Melo and Alberto Sardinha. Ad hoc teamwork by learning teammates'

task. *Autonomous Agents and Multi-Agent Systems*, 30:175–219, 2016. 1, 2.1

[32] Reuth Mirsky, William Macke, Andy Wang, Harel Yedidsion, and Peter Stone. A penny for your thoughts: The value of communication in ad hoc teamwork. International Joint Conference on Artificial Intelligence, 2020. 1, 2.1

[33] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50, 2020. 1

[34] Rupal Nigam, Niket Parikh, Mikihisa Yuasa, and Huy T Tran. Coordination in ad hoc teams with generalized policy improvement. 2.1

[35] Shayegan Omidshafiei, Dong-Ki Kim, Miao Liu, Gerald Tesauro, Matthew Riemer, Christopher Amato, Murray Campbell, and Jonathan P How. Learning to teach in cooperative multiagent reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6128–6136, 2019. 2.1

[36] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020. 2.2

[37] João G Ribeiro, Cassandro Martinho, Alberto Sardinha, and Francisco S Melo. Assisting unknown teammates in unknown tasks: Ad hoc teamwork under partial observability. *arXiv preprint arXiv:2201.03538*, 2022. 1, 2.1

[38] João G Ribeiro, Gonçalo Rodrigues, Alberto Sardinha, and Francisco S Melo. Teamster: Model-based reinforcement learning for ad hoc teamwork. *Artificial Intelligence*, 324:104013, 2023. 1, 2.1, 5

[39] Trevor Sarratt and Arnav Jhala. Policy communication for coordination with unknown teammates. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 2.1

[40] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015. 2.2

[41] Peter Stone, Gal Kaminka, Sarit Kraus, and Jeffrey Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 1504–1509, 2010. 1, 2.1

[42] Zikang Tian, Ruizhi Chen, Xing Hu, Ling Li, Rui Zhang, Fan Wu, Shaohui Peng, Jiaming Guo, Zidong Du, Qi Guo, et al. Decompose a task into generalizable

subtasks in multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024. 2.1

[43] Lisa Torrey and Matthew Taylor. Teaching on a budget: Agents advising agents in reinforcement learning. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1053–1060, 2013. 1, 2.1, 3.3

[44] Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Roma: Multi-agent reinforcement learning with emergent roles. *arXiv preprint arXiv:2003.08039*, 2020. 1, 2.1

[45] Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. Rode: Learning roles to decompose multi-agent tasks. *arXiv preprint arXiv:2010.01523*, 2020. 1, 2.1, 5

[46] Lisheng Wu and Ke Chen. Goal exploration augmentation via pre-trained skills for sparse-reward long-horizon goal-conditioned reinforcement learning. *Machine Learning*, pages 1–31, 2024. 4.2.2

[47] Annie Xie, Dylan Losey, Ryan Tolsma, Chelsea Finn, and Dorsa Sadigh. Learning latent representations to influence multi-agent interaction. In *Conference on robot learning*, pages 575–588. PMLR, 2021. 5

[48] Ju Xu and Zhanxing Zhu. Reinforced continual learning. *Advances in neural information processing systems*, 31, 2018. 4.3.1

[49] Zhenghai Xue, Zhenghao Peng, Quanyi Li, Zhihan Liu, and Bolei Zhou. Guarded policy optimization with imperfect online demonstrations. *arXiv preprint arXiv:2303.01728*, 2023. 3.3

[50] Jiachen Yang, Igor Borovikov, and Hongyuan Zha. Hierarchical cooperative multi-agent reinforcement learning with skill discovery. *arXiv preprint arXiv:1912.03558*, 2019. 2.1, 5

[51] Mingyu Yang, Jian Zhao, Xunhan Hu, Wengang Zhou, Jiangcheng Zhu, and Houqiang Li. Ldsa: Learning dynamic subtask assignment in cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 35:1698–1710, 2022. 2.1

[52] Tianpei Yang, Weixun Wang, Hongyao Tang, Jianye Hao, Zhaopeng Meng, Hangyu Mao, Dong Li, Wulong Liu, Yingfeng Chen, Yujing Hu, et al. An efficient transfer learning framework for multiagent reinforcement learning. *Advances in neural information processing systems*, 34:17037–17048, 2021. 2.1

[53] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624, 2022. 2.2