Implicitly Aligning Humans and Autonomous Agents through Shared Task Abstractions

Stéphane Aroca-Ouellette[1], **Miguel Aroca-Ouellette**[2], **Katharina von der Wense**[1,3] and **Alessandro Roncone**[1]
[1]University of Colorado Boulder
[2]Independent Researcher
[3]Johannes-Gutenberg Universität Mainz
stephane.aroca-ouellette@colorado.edu

Abstract

In collaborative tasks, autonomous agents fall short of humans in their capability to quickly adapt to new and unfamiliar teammates. We posit that a limiting factor for zero-shot coordination is the lack of shared task abstractions, a mechanism humans rely on to implicitly align with teammates. To address this gap, we introduce HA^2: Hierarchical Ad Hoc Agents, a framework leveraging hierarchical reinforcement learning to mimic the structured approach humans use in collaboration. We evaluate HA^2 in the Overcooked environment, demonstrating statistically significant improvement over baselines when paired with both unseen agents and humans, providing better resilience to environmental shifts, and outperforming state-of-the-art methods.

1 Introduction

Successful collaboration requires individuals to efficiently adapt to new teammates. This capability, often referred to as ad hoc teaming [Barrett *et al.*, 2016] or zero-shot coordination [Hu *et al.*, 2020], is an area where humans consistently outperform state-of-the-art autonomous agents. We argue that this disparity arises because humans have access to shared task abstractions [Stanton, 2006], which provide a common foundation that facilitates seamless, implicit coordination. In this paper, we argue that maximizing an agent's ability to collaborate with humans requires providing them with shared task structures and demonstrate the effectiveness of this approach through a large-scale human study.

To elucidate the intricacies and challenges of zero-shot coordination between humans and agents, let's analyze a scenario in the collaborative game "Overcooked", in which players serve as many soups as possible within a time limit. In Fig. 1, the ad hoc agent is playing with an unfamiliar teammate and must decide how to act next. One option, which we define as the '*individual*' strategy, involves obtaining an onion and placing it directly into the pot. This behavior is conservative but suboptimal: it can achieve a moderate score with any player, but will never achieve a top score. Conversely, the agent could opt for a '*coordinated*' strategy, where the blue agent passes onions on the middle counter, hoping their teammate moves

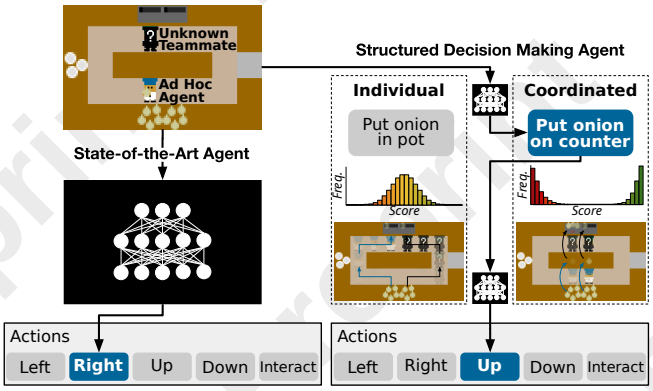

Figure 1: Depicted is a scenario in the Overcooked game where an agent is working with a new teammate. The agent could choose to play a coordinated strategy that is more efficient than an alternative individual strategy, but runs the risk of failure if cooperation is not achieved. Successful ad hoc teaming requires not only being able to perform multiple strategies, but also know when to apply the different strategies. Current SotA approaches subsume these decisions into a single black-box; in contrast, we propose that a structured approach to these decisions provides significant benefits.

them from the counter to a pot. This strategy is more efficient since it eliminates the long walk around the kitchen, but it carries risk as success hinges on both agents adhering to the strategy. This example illustrates a key challenge in zero-shot teaming: an agent must not only *acquire multiple distinct behaviors*, it also needs to *be able to quickly identify which behavior is most suitable for its teammate's skill level.*

Recent works have tried to overcome the challenges of rapidly adapting to new teammates by leveraging teammates of diverse capabilities. Agents have been trained with teammates that emulate human behavior [Carroll *et al.*, 2019] or with a population of teammates that varying levels of proficiency [Strouse *et al.*, 2021; Lucas and Allen, 2022; Lou *et al.*, 2023; Liang *et al.*, 2024]. However, these approaches sidestep a key factor for achieving genuinely collaborative interaction; as depicted in Fig. 1, current state-of-the-art systems consolidate high-level strategy decisions and low-level movement decision into one black box model. In contrast, humans are known to leverage hierarchical frameworks for cognitive processing [Williams, 2022], task management [Annett, 2003;

Zhou, 2013], and human-human coordination [Stanton, 2006], and human-robot collaboration [Roncone *et al.*, 2017; Mangin *et al.*, 2022]. Further, it has been proposed that structured hierarchies are a core component of the human ability for fast generalization [Tenenbaum *et al.*, 2011]. In this paper, we design autonomous agents equipped with hierarchical structures that provide shared task abstractions that enable more efficient alignment with humans. These structures enable agents to focus on the most relevant information for the respective level of abstraction, prevent agents from overfitting to specific training patterns, and create task-oriented agents who may be more understandable to humans. While the benefits of shared task hierarchies are well-established in certain research domains [Ichter *et al.*, 2022; Wang *et al.*, 2024; Annett, 2003], state-of-the-art methods in human-agent interaction [Lou *et al.*, 2023; Liang *et al.*, 2024] have yet to capitalize on this critical concept. This paper addresses this significant gap and advocates that shared task hierarchies should play a central role in human-agent interaction. Our findings show that leveraging shared task hierarchies can provide greater improvements compared to increasing diversity of training agents (cf. Section 5.3).

In all, we present *Hierarchical Ad Hoc Agents* (HA^2), a method that leverages hierarchical reinforcement learning (HRL) to equip an agent with both low-level, efficient maneuvering behaviors and high-level, team-oriented strategies for effective synchronization with human teammates (cf. Fig. 2). Importantly, HA^2 is agnostic to the underlying training algorithms, and serves as an augmentative layer that complements state-of-the-art methods (SotA, e.g., [Strouse *et al.*, 2021; Carroll *et al.*, 2019; Lou *et al.*, 2023; Liang *et al.*, 2024]), leading to statistically significant improvements. Further, this is a deeply generalizable method, as humans have demonstrated the ability to create tasks hierarchies across a broad range of human-human collaborative tasks [Stanton, 2006]. Through extensive evaluations, we find that HA^2offers statistically significant advantages with highlighted by the following contributions: 1) HA^2 outperforms all baselines by over 18.0% when paired with a set of unseen agents, and 2) by over 18.3% when paired with humans. Moreover, 3) HA^2 is significantly preferred by humans, and found to be more fluent, trusted, and cooperative than baselines. To further test the generalizability provided by hierarchical structures, we test the agents zero-shot on modified versions of the game layouts and show that 4) HA^2 is more robust environmental changes, outperforming baselines by more than 10.5x on these layouts. Code is available at https://github.com/HIRO-group/HA2.

2 Related Works

Zero-Shot Coordination The fast-evolving landscape of deep RL agents that interact in the real world has prompted increased investigation into how they can and should interact with humans [Dafoe *et al.*, 2020; Mirsky *et al.*, 2022]. A critical challenge is the development of agents capable of zero-shot coordination with human partners [Stone *et al.*, 2010].

Training partners Prior research identified the limitations of agents trained via self-play—most notably, their behavioral rigidity. To address this, work has enhanced self-play through robust strategy discovery [Hu *et al.*, 2020; Cui *et al.*, 2021; Sarkar *et al.*, 2023], off-belief learning [Hu *et al.*, 2021], or training with a population of pretrained agents [McKee *et al.*, 2022]. Notable advances were made with the use of teammates trained via imitation learning [Carroll *et al.*, 2019], that vary in ability [Strouse *et al.*, 2021], or are specifically trained to be diverse [Lucas and Allen, 2022; Zhao *et al.*, 2023; Lupu *et al.*, 2021]. More recent work has investigated ensembling training partners to create a richer diversity without additional computational costs [Lou *et al.*, 2023]. Our work is directly compatible with this body of work by augmenting the agents with human-aligned structures.

Intention Prediction and Planners A different line of research has focused on modeling the teammate's intention prediction for collaborative tasks [Melo and Sardinha, 2015; Nguyen *et al.*, 2011]. This work often leverages online planners, which have nice properties for ad hoc agents within certain restrictions [Wu *et al.*, 2011]. Although [Wu *et al.*, 2021; Pöppel *et al.*, 2022] employ hierarchical structures in their planners, they lack real-world applicability due to computational constraints for complex environments and have not been tested with real human game-play. [Carroll *et al.*, 2019] compared their models to planning-based methods, but were only able to use planners in two of their fivelayouts. When playing any agent for which they did not have an accurate model of (e.g., a human), performance dropped dramatically. Though out-of-scope here, we believe that modeling would compliment our proposed method.

Type-based Agents Ad hoc teaming has also been investigated by using type-based agents that rely on a pre-generated population of diverse teammates. The PLASTIC framework [Barrett *et al.*, 2016] offers two strategies: PLASTIC-Model, which employs the most human-like teammate for action planning, and PLASTIC-Policy, which first learns then selects the most appropriate complementary policy for each teammate. This latter approach is paralleled in [Li *et al.*, 2021], albeit with a distinct similarity metric. Finally, [Chen *et al.*, 2020] takes this further by subsuming teammates into world models, and using them to learn respective policies.

Hierarchical Reinforcement Learning As breaking down complex tasks into sub-tasks is used in many facets of life, HRL is a well-studied area [Sutton *et al.*, 1999; Dietterich, 2000; Dayan and Hinton, 1992; Vezhnevets *et al.*, 2017]. HRL has also been extended to multi-agent cooperation, either by deploying a central manager to oversee multiple agents [Ahilan and Dayan, 2019], or by imbuing each agent with its own hierarchical architecture [Ghavamzadeh *et al.*, 2006; Makar *et al.*, 2001]. Other work has ventured into learning the sub-tasks [Yang *et al.*, 2023; Wang *et al.*, 2021]. Most similar to our work is HiPT [Loo *et al.*, 2023]. However, the work differs in several critical ways: 1) Our motivation stems from aligning structures between humans and agents; thus, our abstracted layer between Worker and Manager is fully human interpretable. 2) Our method consistently outperform HiPT across all layouts. See Section 5.3 for comparative results. 3) We show that our architecture enables greater generalizeability to shifts in the game layouts, a feature not shown in HiPT. 4) We show that HA^2 provides significant benefits regardless of which training teammates are used.

3 Method

3.1 Environment

Following prior work in zero-shot human-AI teaming [Carroll *et al.*, 2019; Strouse *et al.*, 2021; Aroca-Ouellette *et al.*, 2023], we study the use of hierarchical structures using all five layouts in the Overcooked environment developed by [Carroll *et al.*, 2019]. The goal of this collaborative game is to serve as many soups as possible in the time limit. To accomplish this, players must perform a sequence of task from retrieving onions and placing them in a pot to serving completed soups. Upon service, the team is rewarded with 20 points.

At each timestep, each player can choose to move {*up, down, left, right*}, *interact* with an object (for picking up/placing/serving objects), or *stay* still. To effectively play Overcooked, agents must both coordinate on high-level sub-tasks and low-level movement patterns. At the sub-task level, players should avoid redundant and inefficient sub-tasks such as each retrieving a dish if only one soup is cooking. At a low-level, players must be cautious to avoid collisions. This layered complexity makes Overcooked a particularly apt testbed for human-agent collaboration.

3.2 Sub-tasks

In human-human game-play, synchronization typically occurs at this sub-task level. In the Overcooked environment, sub-task identification is facilitated by the *interact* action, which serves as a delineating event. Utilizing it, we enumerate all possible outcomes resulting from the 'interact' action to define our set of sub-tasks: (1) Pick up onion from onion dispenser (2) Pick up onion from counter (3) Pick up dish from dish dispenser (4) Pick up dish from counter (5) Place onion in pot (6) Place onion on counter (7) Get soup from pot (8) Place dish on counter (9) Get soup from pot (10) Place soup on counter (11) Serve soup (12) Unknown

3.3 HA^2: Hierarchical Ad Hoc Agent

Inspired by the notion that cognitive and behavioral alignment between humans and agents enhances human-AI Teaming, we adapt FuN [Vezhnevets *et al.*, 2017] to introduce HA^2: Hierarchical Ad Hoc Agents. HA^2 aims to approach human-agent teaming as one would approach human-human teaming: by developing a shared and mutually understandable task hierarchy. HA^2 (cf. Fig. 2) consists of two tiers of models: a Worker, that focuses on the efficient execution of sub-tasks while avoiding collisions; and a Manager that focuses on high-level-task synchronization with its teammate. This decoupled architecture not only facilitates collaboration by allowing the models to focus on the information at their level of abstraction, but it additionally streamlines both their learning processes.

The Worker is tasked with learning how to complete sub-tasks. With reference to Section 3.2, this consists of moving to a certain location with a specific orientation and *interact*ing with the environment. With this in mind, we add a layer to the lossless observation developed by [Carroll *et al.*, 2019] that indicates the end locations of the current sub-task. For example, for the sub-task 'put onion in pot', each non-full pot would be marked in this layer of the observation. We then create a modified versions of the original environment. In this

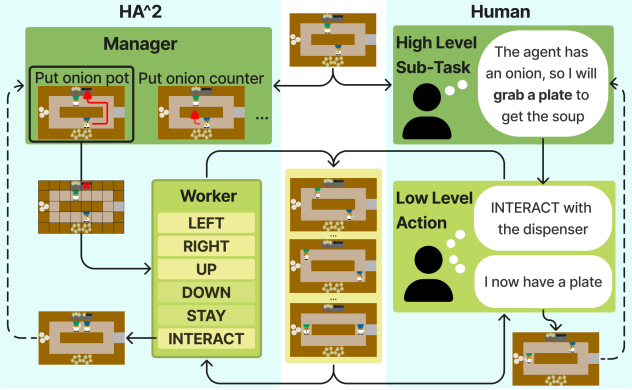

Figure 2: An overview of the HA^2 architecture. Similar to human behavior, an observation is initially processed by the Manager to decide on the next high-level sub-task. Subsequently, the Worker executes the necessary low-level actions to complete the sub-task.

environment, each episode is associated with a sub-task and runs until the agent performs the *interact* action or times out. If the agent completes the correct sub-task, they receive a reward of +1, otherwise they receive a reward of -1. Certain sub-tasks offer additional small rewards for more optimal completion methods. For tasks involving placing objects on or picking up object from counters, an additional reward is added for the numbers of steps that can be saved by using that counter compared to moving from the agent's current location. For placing onions in pots there is an additional reward for placing onions in the pot that has more onions. When an episode ends, a new sub-task is sampled from the list of possible sub-tasks given the current state. The sub-tasks are sampled inversely proportionally to how often that sub-task has been used previously in training to get a more even coverage of sub-tasks during training. Once the horizon of the original environment is reached, the environment resets the state to the standard start state and a new sub-task is sampled. Due to its undefined nature, we omit the unknown sub-task at this stage.

The Manager is responsible for deciding which sub-task should be completed next. Specifically, it is trained to output a distribution over sub-tasks. To train the Manager, we again create a modified version of the original environment. In this environment, the action space is one of the 12 possible sub-tasks. If the manager selects the undefined sub-task, the additional observation layer passed to the worker is empty. Unlike the base environment, not all actions are possible for each state: for example, the agent cannot put an onion in the pot if they are not carrying an onion. To address this, we mask out all sub-tasks which are not possible at the current time-step. Once the sub-task has been chosen, the associated observation is passed to the Worker, who selects the low-level action for that timestep. We found that having the Manager select sub-tasks at each time-step improved sample efficiency and overall performance given the computational budget. The reward structure is the same as the base environment with a reward of 20 for each soup served.

4 Experimental Design

4.1 Baselines and HA^2 models

We implement two baselines representative of the existing approaches in the field: Behavioral Cloning Play (BCP), [Carroll *et al.*, 2019] and Fictitious Co-Play (FCP) [Strouse *et al.*, 2021]. BCP[1] was designed to have an RL agent learn how to play with the movement patterns of a human. To do this, a behavioral cloning (BC) model is first trained from human data, and then a RL agent is trained with the BC model as its teammate. FCP is designed to have agents learn to play with a wide range of teammates. It first learns a population of self-play agents who vary in architecture and seed. It then augments the population by using three versions of each agent: its random initialization, roughly midway through training, and after completing training. It then trains the FCP agent to play with the whole population of agents.

We note that HA^2 serves as an architectural enhancement within the agent, and that the agent can be trained using any type of teammate. To demonstrate the applicability of HA^2 to existing methods, we train two versions of HA^2. HA$^2{}_{BCP}$ is trained using a BC teammate and is directly comparable to BCP. HA$^2{}_{FCP}$ is trained using a FCP population and is directly comparable to FCP. We train five iterations of each of the four agents using different random seeds and report the mean and standard error across seeds.

To train the BC models, we closely follow the implementation in [Carroll *et al.*, 2019], using their feature encoding as observation as well as their provided data. We make two small changes which we found improves performance. First, we remove all time-steps where both agents perform the *stay* action. Second, in the loss, we weigh each action inversely proportional to their frequency in the dataset. Following [Carroll *et al.*, 2019], We divide the data in half, and train two models. The better model is used as the human proxy, and the worse model as the BC model. We note that these two agents are the only agents where we train one model per layout.

The RL agents train one model for all layouts and use the 7x7 egocentric view developed by [Strouse *et al.*, 2021]. However, instead of the convolutional neural network (CNN) used in [Strouse *et al.*, 2021], we flatten the observation and pass it through a two-layer multilayer perceptron (MLP) as we found it outperforms a CNN. We experiment using recurrent networks, as in [Strouse *et al.*, 2021], but found they also underperform MLPs. We additionally experiment with frame stacking, which we found outperforms a Recurrent PPO, but underperforms the standard PPO approach.

The training population for the FCP agent and HA$^2{}_{FCP}$ consists of eight self-play agents that vary in seed, hidden dimension (64 and 256), and whether or not they use frame stacking. When training the population, we found that agents learned on the different layouts at different rates. To maintain a good balance of skill levels for each layout, we use different middle checkpoints for each population agent for each layout, with the checkpoints corresponding to points closest to where the agent reaches half the highest score for that layout.

[1]BCP was originally named PPO$_{BC}$ in [Carroll *et al.*, 2019] and renamed by [Strouse *et al.*, 2021] to BCP. We use BCP in this paper for succinctness.

Each population agent was trained for 10 million in-game steps and the BC agents were trained for 300 epochs. HA^2 and the baselines train at different rates with HA^2 taking the longest to train since it requires two predictions — one from the manager, the other from the worker — at each timestep. To keep a fair comparison, we train each agent for 48 hours using the same V100 GPU. For HA^2, we use 24 hours for the Worker and 24 hours for the Manager. The 48 hours equate to $\sim$119 million timesteps for BCP, $\sim$119 million timesteps for FCP, and $\sim$66 million timesteps for HA^2 (31 million for the Worker and 35 million for the Manager). We note that all agents reached over 98% of their top performance within the first half of this training.

4.2 Research Questions and Experiments

RQ1: Does HA^2 improve performance with unseen agents? We hypothesize that the addition of a hierarchical structure will help the agent's models focus more closely to the salient information at their respective level of abstraction. Further, we hypothesize that it learn more general game concepts by preventing it from over-fitting to any specific training patterns. Since the reward is fully shared and because the agent can impact its training teammate's actions by influencing the observations, it follows that the agent will also maximize its actions to promote its teammates' high-scoring behaviors. When using low-level actions, this can quickly lead to weird specificities that generalize poorly—e.g. waiting to put the onion in the pot until the teammate is in a specific spot and facing a specific way. Enforcing a hierarchical structure should mitigates this effect since the Worker is not rewarded by teammate behaviors and the Manager has no control over the movement of the Worker.

To test this initial hypothesis, we compare the performance of HA$^2{}_{BCP}$ and HA$^2{}_{FCP}$ to their respective baselines when paired with three agents of varying capability: a self-play model (fully distinct from any in the FCP population), the human proxy model, and an agent that performs random actions. See Section 5.1 for the results of this experiment.

RQ2: Does HA^2 create higher performing and more fluent human-agent teams? The primary motivation for this work is to develop agents that are effective at collaborating with humans. Human teammates present unique challenges to autonomous agents—prime among them the fact that humans have a significantly higher ability to adapt. In turn, this requires agents paired with humans to not only being able to adapt themselves, but also to make it easy for a teammate to adapt to them. Beyond the improved generalizability we test for, we hypothesize that HA^2's structure will make them more task-focused and in turn more understandable to humans.

To this end, we conduct an IRB-approved online user study. We use a within-subjects design for the study where each participant plays with two agents on each layout. To test our above hypothesis, we evaluate both objective performance and subjective preferences between pairs of agents. Each participant was first provided with an instruction page, before completing a short tutorial that required them to complete all the steps to serve a soup to move on. Each participant then played an 80 second round (400 steps at 5FPS) with each agent on one of the layouts. Between each round, the participants

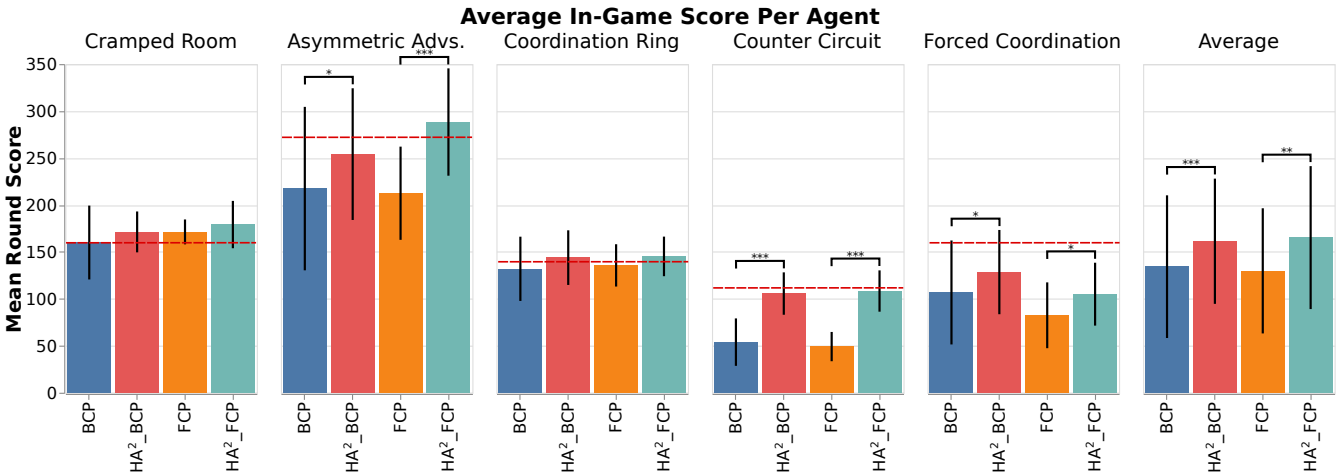

Figure 3: Average score of HA^{2}s and the BCP and FCP baselines when paired with humans on each of the layouts. Each round was 80 seconds long at 5 FPS (T=400 steps). Significance markers: *=p$<$ 0.05, **=p$<$ 0.005, ***=p$<$ 0.0005. The red line indicates the max human-human score achieved on that layout from [Carroll *et al.*, 2019] normalized to 400 steps.

had to answer eight questions adapted from [Hoffman, 2019] asking them how much they agreed or disagreed with statement on a 7-point likert scale. After each pair, they were asked to rank which of the two agents they preferred. They then repeated this process for the other four layouts. The order of the layouts and agent they played with first within each layout was randomized. The chef that the agent and human controlled were consistent between the two comparative agents, but randomized between layouts and participants.

For this research question, we run two pairwise comparisons: HA$^2{}_{BCP}$ vs. BCP and HA$^2{}_{FCP}$ vs. FCP. We recruit 50 participants for the BCP comparison and 25 participants for FCP comparison. We filter out any participant that did not complete the full trial. We additionally filter out any pair of rounds (i.e., comparing two agents on one layout) where the human performed fewer than five subtasks in either round. This leaves us with 47 and 24 participants respectively. We recruit all participants from prolific.com. Participants were compensated using a base rate of $3.00 plus a bonus incentive of $0.04 for each dish served. The average participant compensation for these two studies was $15.79/hour. This human survey was approved an Institutional Review Board, indicating that it presented minimal risk to participants. All participants provided informed consent for the study. Results for these human studies are in Section 5.2.

RQ3: Can HA^2 agents generalize better to changes in the layouts? Since the hierarchical structure we are using is intrinsic to Overcooked at large, we posit that HA^2 should not only generalize better to different agents, but also generalize better to shifts in the layouts. For this experiment, we create a modified version of each layout by swapping two tiles in each layout. Since we do not have any trained unseen agents on these layouts, we evaluate the HA^{2}s and their respective baselines on these modified layouts by teaming each agent with themselves. Section 5.4 shows results for this experiment.

	BCP	HA$^2{}_{BCP}$	FCP	HA$^2{}_{FCP}$
AA	$199.9_{\pm 8.0}$	$278.3_{\pm 6.3}$	$210.8_{\pm 40.0}$	$\mathbf{293.5}_{\pm 7.2}$
CoR	$79.2_{\pm 4.2}$	$133.3_{\pm 3.2}$	$138.6_{\pm 2.5}$	$\mathbf{147.6}_{\pm 0.8}$
CC	$17.1_{\pm 11.4}$	$91.2_{\pm 5.0}$	$74.3_{\pm 19.3}$	$\mathbf{99.9}_{\pm 2.8}$
CrR	$143.1_{\pm 13.8}$	$177.7_{\pm 4.1}$	$183.9_{\pm 4.7}$	$\mathbf{185.5}_{\pm 2.3}$
FC	$73.1_{\pm 5.6}$	$\mathbf{77.6}_{\pm 3.5}$	$56.7_{\pm 4.1}$	$58.4_{\pm 4.8}$
Avg.	$102.5_{\pm 4.5}$	$151.6_{\pm 2.4}$	$133.0_{\pm 8.8}$	$\mathbf{157.0}_{\pm 1.3}$
$\sim$ AA	$23.6_{\pm 41.5}$	$157.2_{\pm 40.4}$	$7.6_{\pm 14.2}$	$\mathbf{208.0}_{\pm 28.1}$
$\sim$ CoR	$11.6_{\pm 11.4}$	$\mathbf{152.8}_{\pm 7.0}$	$22.8_{\pm 6.4}$	$143.2_{\pm 12.6}$
$\sim$ CC	$2.0_{\pm 2.5}$	$70.0_{\pm 15.8}$	$9.2_{\pm 14.5}$	$\mathbf{110.0}_{\pm 35.5}$
$\sim$ CrR	$5.6_{\pm 2.9}$	$\mathbf{162.4}_{\pm 15.2}$	$0.8_{\pm 1.6}$	$154.8_{\pm 36.8}$
$\sim$ FC	$10.4_{\pm 8.9}$	$17.2_{\pm 31.5}$	$3.2_{\pm 3.0}$	$\mathbf{20.8}_{\pm 31.7}$
$\sim$ Avg.	$10.6_{\pm 9.5}$	$111.9_{\pm 13.4}$	$8.7_{\pm 2.4}$	$\mathbf{127.3}_{\pm 7.1}$

Table 1: Mean$_{\pm \text{SE}}$ score across 5 random training seeds for HA^{2}s and their respective baselines. The score of each trained agent is the average across 10 trials of T=400 steps with each teammate. In the original layouts, the teammates are an unseen self-play agent, the human proxy, and a random agent. In the modified layouts (denoted with $\sim$), the teammate is a copy of the acting agent.

4.3 Significance Testing

For each pairwise comparison, we perform t-tests to measure significance. For the significance of team performance, we compare the score achieved directly. For the ranking significance, we mapped every instance where an agent was preferred over its counterpart to a score of 1 and every other instance to a score of 0. We then used these scores to perform the t-tests. For the Likert questions, we mapped each agreement level to a score between -3 (strong disagree) and 3 (strong agree), with the neutral score being 0. We normalize all participants scores to have a mean of 0 and then use these score for perform the t-tests.

5 Results

5.1 Zero-shot Coordination with Unseen Agents

We first compare HA^2 to the baselines—BCP and FCP—on their ability to generalize to new unseen agents. The results in Table 1 clearly demonstrate the improvement provided by the hierarchical structure, with the HA^{2}s outperforming their respective baselines on every layout. Using HA^2 afforded an improvement of 47.9% when using BCP, and an improvement of 18.0% when using FCP. HA$^2{}_{BCP}$ performs best on forced coordination, and HA$^2{}_{FCP}$ performs best on all the other layouts and overall. We discuss a possible cause of this in Section 5.4. We additionally note that HA^2 is more robust to the random seed than the baselines, with a lower standard error on each layout across the 5 random seeds.

5.2 Zero-shot Coordination with Humans

We now present the findings of our human study comparing HA^2 and the baselines. Results in Fig. 3 demonstrate that in both HA$^2{}_{BCP}$ and HA$^2{}_{FCP}$ significantly outperform their respective counterparts on the overall score achieved, and on the asymmetric advantages, counter circuit, and forced coordination layouts. We note that the scores of the baselines in cramped room and coordination ring are closer to the maximum human-human [2] score achieved (dotted red line), leaving less room for improvement. As such, we anticipated there would be a smaller variability on these layouts. Table 2 further shows that HA^2 was significantly preferred over their counterparts. In RQ3, we had hypothesized that HA^{2}s would improve human-agent teaming because they are easier to understand, and therefore easier to adapt to. Fig. 4 supports this hypothesis and shows that in both comparisons of HA^2 to the baselines, humans rated the HA^{2}s as significantly more understandable, intelligent, and cooperative. In the case of FCP and HA$^2{}_{FCP}$, humans also found that HA^2 was significantly more fluent, trusted, and more helpful at helping the humans adapt to the task. These results strongly support using shared task hierarchies for human-agent collaboration.

In line with the results with unseen agents, forced coordination is the one layout where BCP and HA$^2{}_{BCP}$ outperform their FCP counter parts. We hypothesize that this is due to it being the only layout where having an untrained teammate blocks the agent's ability to earn a reward. Since a third of FCP's training population are untrained agents, FCP and HA$^2{}_{FCP}$ effectively lose a third of their training. The results in the appendix of [Strouse *et al.*, 2021] support this hypothesis showing that forced coordination is least benefitted by FCP. This can likely be remedied by excluding the untrained partners in layouts where coordination is required to achieve a non-zero score.

	% Preferred	p-value
HA$^2{}_{BCP}$ over BCP	57.68	0.0070
HA$^2{}_{FCP}$ over FCP	65.25	0.0000018

Table 2: Human preference between pairs of agents and their respective significance.

[2]From [Carroll *et al.*, 2019]'s data normalized to 400 timesteps.

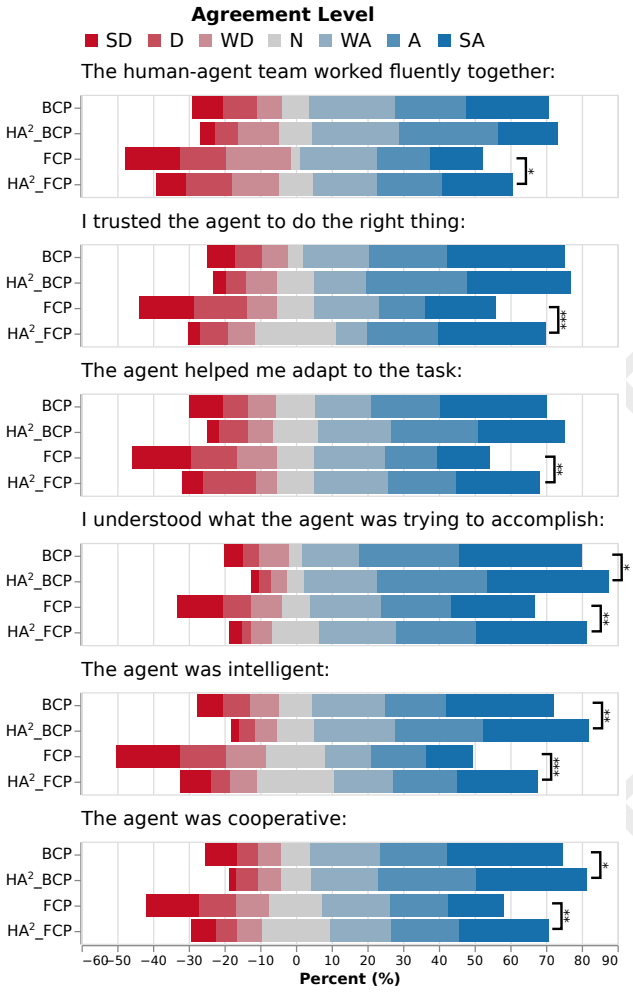

Figure 4: Subset of results from the eight Likert-scale questions that participants answer after playing with each agent for the comparison between HA^2 and their baselines. Bars that are more blue indicate that people agree more strongly with the statement. Conversely, more red indicates that people disagreed more strongly with the statement. Significance markers: *=p$<$ 0.05, **=p$<$ 0.005, ***=p$<$ 0.0005. Legend: SD=Strongly Disagree, D=Disagree, WD=Weakly Disagree, N=Neutral, WA=Weakly Agree, A=Agree, SA=Strongly Agree.

Interestingly, when analyzing Figs. 3 and 4, we noticed that even if their overall scores were generally worse, BCP and HA$^2{}_{BCP}$ were better perceived on every subjective metric relative to their FCP and HA$^2{}_{FCP}$. This does pose the question of whether utilizing human behavior in training does provide a more human-like game-play, and in turn a more fluid experience for humans, which is supported by the results in [Liang *et al.*, 2024]. We leave a more thorough investigation of this question, as well as the relationship between team performance and human perception, to future work.

5.3 Comparison to State-of-the-Art

In Table 3, we compare HA^2 to results published in other peer-reviewed work that uses the same overcooked environment. As each method employs a range of design decisions,

	Training Steps	W. Proxy	W. Humans
FCP	1.0e9	**157**	119
MEP	5.5e7*	98	98
TrajeDi	5.5e7*	76	87
PECAN	NR	105	134
HiPT	1.0e9	134	131
GAMMA	1.5e8	132	NR
$HA^2{}_{FCP}$	6.6e7	**157**	**165**

Table 3: Results comparing HA^2 to other published results. All results are taken from the respective works and adjusted to 400 timesteps, except for TrajeDi's results which are taken from [Zhao *et al.*, 2023]. NR=not reported. * indicates that separate agents are trained for each layout and that the cumulative step count across layouts is presented. FCP [Strouse *et al.*, 2021], MEP [Zhao *et al.*, 2023], TrajeDi [Lupu *et al.*, 2021], PECAN [Lou *et al.*, 2023], HiPT [Loo *et al.*, 2023], GAMMA [Liang *et al.*, 2024],

this table should be viewed as a comparison of systems. Notably, when paired with a human proxy, HA^2 is tied as the best performing agent, whereas when HA^2 is paired with real humans, HA^2 outperforms all other work by more than 23%, showcasing HA^2's adeptness at human collaboration. This is further emphasized when comparing to the most similar work of HiPT. HA^2 outperforms HiPT by 17% when paired with a human proxy, and by 26% when paired with real humans while using 15.1 times fewer timesteps (1 billion timesteps for HiPT vs ∼66 million timesteps for HA^2). This highlights HA^2's greatest distinction from HiPT: *using* ***human-aligned*** *structures improves the training efficiency and performance of autonomous agents that collaborate with humans.* Lastly, we compare HA^2 to the most recent SotA method: GAMMA [Liang *et al.*, 2024]. Even with a simpler training population, HA^2 outperforms GAMMA by 25% with a proxy human. Further, when paired with real humans on counter circuit, which is the only original layout on which they provide results with real humans, HA^2 outperforms the best version of GAMMA with a score of 110 compared to 91. In all, we show that shared task structures are a critical component when developing collaborative agents.

5.4 Generalization to Shifts in Layouts

We report our results on the generalization ability of HA^2 and the baselines on the altered layouts. The latter half of Table 1 shows that BCP and FCP overfit to the specific layouts and their performance drops dramatically when the layouts are changed. In contrast, the HA^2s are able to maintain a reasonable performance, and are over 10.5x better on the modified layouts. Together with the results in Section 5.1, these results provide strong support for our hypothesis that the hierarchical structure enables the model to learn more generalizable concepts about collaboration and game-play.

6 Discussion

6.1 Summary

This paper offers a comprehensive investigation into the efficacy of leveraging shared task abstractions for enhancing human-agent collaborative systems. The experiments we conducted demonstrate that our Hierarchical Ad Hoc Agent (HA^2) significantly outpaces existing baselines in both quantitative and qualitative assessments. In interactions with human participants, HA^2 created higher performing teams and was perceived by the human as more fluent, more understandable, more cooperative, and more intelligent. Importantly, HA^2 displays robust generalization capabilities not only across diverse agent types but also in response to variations in environmental layouts—outperforming baseline models substantially in these regards. Finally, we highlight how HA^2, even when trained using simpler training partners, outperforms all existing methods when paired with real humans. These findings collectively highlight the utility of human-interpretable hierarchical structures in designing AI agents that are both resilient to changing conditions and intuitively collaborative *on human terms.* We posit that these advancements form a crucial building block toward more performant and efficient human-AI teams.

6.2 Limitations

We now discuss the limitations of our proposed method. The hierarchical structure in HA^2 necessitates additional engineering effort, both in the development of the structure, and the adjustments to the environment required to train the different modules. We note that the method in which to break-down large tasks into sub-tasks to create a hierarchy is not the focus of this work, and has been extensively explored in many domains including human factors research [Annett, 2003; Stanton, 2006], robotics [Ichter *et al.*, 2022], and single-agent long-horizon tasks [Wang *et al.*, 2024]. Rather, the focus of this work is demonstrating the importance of shared task hierarchies in human-agent collaboration.

6.3 Future Work

We envision the following avenues for future work:

First, to incorporate explicit mental models of teammate sub-tasks into agent planning, similar to [Melo and Sardinha, 2015; Nguyen *et al.*, 2011]. We envision that these mental models will synergize with the abstracted manager sub-tasks allowing for more efficient computation of these models, and in turn providing the manager with an efficient understanding of human team members' capabilities and intentions.

Second, we believe HA^2 shows promise as a framework to investigate human-agent communication in collaborative games; it is much easier to communicate at sub-task-level than at action-level.

Ethical Statement

We have shown that leveraging hierarchical structures can yield agents that collaborate more fluently and coherently with humans, potentially fostering greater trust and operational efficiency in human-AI teams. However, it's crucial to recognize that, despite its improved transparency, the HA^2 paradigm is still underpinned by neural networks, which remain inherently opaque. This opacity could induce misplaced trust in the system's capabilities or intentions. Additionally, the increased controllability of the agent through sub-task biasing introduces the potential for misuse, particularly by malicious actors aiming to compromise human safety.

Acknowledgments

This work was supported by the Army Research Laboratory (Grants W911NF-21-2-02905, W911NF-21-2-0126) and by the Office of Naval Research (Grant N00014-22-1-2482).

References

[Ahilan and Dayan, 2019] Sanjeevan Ahilan and Peter Dayan. Feudal multi-agent hierarchies for cooperative reinforcement learning, 2019.

[Annett, 2003] John Annett. Hierarchical task analysis. *Handbook of cognitive task design*, 2:17–35, 2003.

[Aroca-Ouellette *et al.*, 2023] Stéphane Aroca-Ouellette, Miguel Aroca-Ouellette, Upasana Biswas, Katharina Kann, and Alessandro Roncone. Hierarchical reinforcement learning for ad hoc teaming. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, page 2337–2339, 2023.

[Barrett *et al.*, 2016] Samuel Barrett, Avi Rosenfeld, Sarit Kraus, and Peter Stone. Making friends on the fly: Cooperating with new teammates. *Artificial Intelligence*, October 2016.

[Carroll *et al.*, 2019] 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. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[Chen *et al.*, 2020] Shuo Chen, Ewa Andrejczuk, Zhiguang Cao, and Jie Zhang. Aateam: Achieving the ad hoc teamwork by employing the attention mechanism. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7095–7102, Apr. 2020.

[Cui *et al.*, 2021] Brandon Cui, Hengyuan Hu, Luis Pineda, and Jakob Foerster. K-level reasoning for zero-shot coordination in hanabi. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8215–8228. Curran Associates, Inc., 2021.

[Dafoe *et al.*, 2020] Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R. McKee, Joel Z. Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative ai, 2020.

[Dayan and Hinton, 1992] Peter Dayan and Geoffrey E Hinton. Feudal reinforcement learning. In S. Hanson, J. Cowan, and C. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann, 1992.

[Dietterich, 2000] Thomas G. Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *J. Artif. Int. Res.*, 13(1):227–303, nov 2000.

[Ghavamzadeh *et al.*, 2006] Mohammad Ghavamzadeh, Sridhar Mahadevan, and Rajbala Makar. Hierarchical multi-agent reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 13(2):197–229, sep 2006.

[Hoffman, 2019] Guy Hoffman. Evaluating fluency in human–robot collaboration. *IEEE Transactions on Human-Machine Systems*, 49(3):209–218, 2019.

[Hu *et al.*, 2020] Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "Other-play" for zero-shot coordination. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4399–4410. PMLR, 13–18 Jul 2020.

[Hu *et al.*, 2021] Hengyuan Hu, Adam Lerer, Brandon Cui, Luis Pineda, Noam Brown, and Jakob Foerster. Off-belief learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4369–4379. PMLR, 18–24 Jul 2021.

[Ichter *et al.*, 2022] Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, ..., Andy Zeng, Chuyuan Kelly Fu Do as i can, not as i say: Grounding language in robotic affordances. In *6th Annual Conference on Robot Learning*, 2022.

[Li *et al.*, 2021] Huao Li, Tianwei Ni, Siddharth Agrawal, Fan Jia, Suhas Raja, Yikang Gui, Dana Hughes, Michael Lewis, and Katia Sycara. Individualized mutual adaptation in human-agent teams. *IEEE Transactions on Human-Machine Systems*, 51(6):706–714, 2021.

[Liang *et al.*, 2024] Yancheng Liang, Daphne Chen, Abhishek Gupta, Simon Shaolei Du, and Natasha Jaques. Learning to cooperate with humans using generative agents. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[Loo *et al.*, 2023] Yi Loo, Chen Gong, and Malika Meghjani. A hierarchical approach to population training for human-ai collaboration. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 3011–3019. ijcai.org, 2023.

[Lou *et al.*, 2023] Xingzhou Lou, Jiaxian Guo, Junge Zhang, Jun Wang, Kaiqi Huang, and Yali Du. Pecan: Leveraging policy ensemble for context-aware zero-shot human-ai coordination. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, page 679–688, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems.

[Lucas and Allen, 2022] Keane Lucas and Ross E. Allen. Any-play: An intrinsic augmentation for zero-shot coordination. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, page 853–861, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems.

[Lupu *et al.*, 2021] Andrei Lupu, Brandon Cui, Hengyuan Hu, and Jakob Foerster. Trajectory diversity for zero-shot coordination. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine*

Learning Research, pages 7204–7213. PMLR, 18–24 Jul 2021.

[Makar *et al.*, 2001] Rajbala Makar, Sridhar Mahadevan, and Mohammad Ghavamzadeh. Hierarchical multi-agent reinforcement learning. In *Proceedings of the Fifth International Conference on Autonomous Agents*, AGENTS '01, page 246–253, New York, NY, USA, 2001. Association for Computing Machinery.

[Mangin *et al.*, 2022] Olivier Mangin, Alessandro Roncone, and Brian Scassellati. How to be helpful? supportive behaviors and personalization for human-robot collaboration. *Frontiers in Robotics and AI*, 8:725780, 2022.

[McKee *et al.*, 2022] Kevin R. McKee, Joel Z. Leibo, Charlie Beattie, and Richard Everett. Quantifying the effects of environment and population diversity in multi-agent reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 36(1), mar 2022.

[Melo and Sardinha, 2015] Francisco Melo and Alberto Sardinha. Ad hoc teamwork by learning teammates' task. *Autonomous Agents and Multi-Agent Systems*, 30, 01 2015.

[Mirsky *et al.*, 2022] Reuth Mirsky, Ignacio Carlucho, Arrasy Rahman, Elliot Fosong, William Macke, Mohan Sridharan, Peter Stone, and Stefano V. Albrecht. A survey of ad hoc teamwork research. In Dorothea Baumeister and Jörg Rothe, editors, *Multi-Agent Systems*, pages 275–293, Cham, 2022. Springer International Publishing.

[Nguyen *et al.*, 2011] Truong-Huy Nguyen, David Hsu, Wee-Sun Lee, Tze-Yun Leong, Leslie Kaelbling, Tomas Lozano-Perez, and Andrew Grant. Capir: Collaborative action planning with intention recognition. 10 2011.

[Pöppel *et al.*, 2022] Jan Pöppel, Sebastian Kahl, and Stefan Kopp. Resonating minds - emergent collaboration through hierarchical active inference. *Cogn. Comput.*, 14(2):581–601, 2022.

[Roncone *et al.*, 2017] Alessandro Roncone, Olivier Mangin, and Brian Scassellati. Transparent role assignment and task allocation in human robot collaboration. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1014–1021. IEEE, 2017.

[Sarkar *et al.*, 2023] Bidipta Sarkar, Andy Shih, and Dorsa Sadigh. Diverse conventions for human-AI collaboration. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[Stanton, 2006] Neville A Stanton. Hierarchical task analysis: Developments, applications, and extensions. *Applied ergonomics*, 37(1):55–79, 2006.

[Stone *et al.*, 2010] Peter Stone, Gal A. Kaminka, Sarit Kraus, and Jeffrey S. Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence*, July 2010.

[Strouse *et al.*, 2021] DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 14502–14515. Curran Associates, Inc., 2021.

[Sutton *et al.*, 1999] Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181–211, 1999.

[Tenenbaum *et al.*, 2011] Joshua B. Tenenbaum, Charles Kemp, Thomas L. Griffiths, and Noah D. Goodman. How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022):1279–1285, 2011.

[Vezhnevets *et al.*, 2017] Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. FeUdal networks for hierarchical reinforcement learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3540–3549. PMLR, 06–11 Aug 2017.

[Wang *et al.*, 2021] Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. {RODE}: Learning roles to decompose multi-agent tasks. In *International Conference on Learning Representations*, 2021.

[Wang *et al.*, 2024] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024.

[Williams, 2022] Gwydion Williams. *Hierarchical Influences on Human Decision-Making*. PhD thesis, UCL (University College London), 2022.

[Wu *et al.*, 2011] Feng Wu, Shlomo Zilberstein, and Xiaoping Chen. Online planning for ad hoc autonomous agent teams. pages 439–445, 01 2011.

[Wu *et al.*, 2021] Sarah A. Wu, Rose E. Wang, James A. Evans, Joshua B. Tenenbaum, David C. Parkes, and Max Kleiman-Weiner. Too many cooks: Coordinating multi-agent collaboration through inverse planning. *Topics in Cognitive Science*, n/a(n/a), 2021.

[Yang *et al.*, 2023] Mingyu Yang, Yaodong Yang, Zhenbo Lu, Wengang Zhou, and Houqiang Li. Hierarchical multi-agent skill discovery. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[Zhao *et al.*, 2023] Rui Zhao, Jinming Song, Yufeng Yuan, Haifeng Hu, Yang Gao, Yi Wu, Zhongqian Sun, and Wei Yang. Maximum entropy population-based training for zero-shot human-ai coordination. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(5):6145–6153, Jun. 2023.

[Zhou, 2013] Yue Maggie Zhou. Designing for complexity: Using divisions and hierarchy to manage complex tasks. *Organization Science*, 24(2):339–355, 2013.