Adapting to Unknown Conventions in Cooperative Multi-Agent RL

CS 330 Fall 2022 - Final Report

Bidipta Sarkar

ABSTRACT

In multi-agent cooperative games, establishing a shared convention with teammates is crucial for seamless coordination. Unfortunately, the standard multi-agent reinforcement learning technique of self-play results in agents that are only aware of a single type of convention and therefore do not generalize well to other partners. In this project, we study techniques for adapting to unknown conventions with only a few episodes of online interaction. We start by analyzing the problem of few-shot coordination as a single-agent multi-task problem. In particular, we notice that the only difference between conventions from a single-agent perspective is the transition function, which is induced from the partner's policy. To generate the set of tasks, we want an algorithm that creates a diverse set of conventions. We use the CoMeDi algorithm (which I started developing prior to this project) to generate a sequence of 8 conventions that perform well individually but have a low cross-play score when playing with a different partner. We use the first 7 conventions as the training set for our techniques and hold out the last convention as our test task.

Using the diverse training set, we construct a common best response agent that tries to maximize its expected reward with each of the training partners. We observe that this best response agent is able to extrapolate beyond its training set, resulting in a higher score with the held out convention than any agent in the training set. However, its performance is still lower than that of a human player, so we investigate techniques to improve performance in the few-shot setting. Although we were not able to notice significant performance improvements using MAML, we did see that few-shot fine tuning from the best response allowed the agent to improve significantly, reaching human-level performance with just a dozen episodes of online play with the held-out partner. We also notice that having a larger diverse training set improves few-shot performance in addition to improving zero-shot scores.

From our experiments, we conclude that fine-tuning a common best response agent to a diverse set of conventions enables fast adaptation to unknown conventions. This procedure can allow us to accelerate the process of human-AI adaptation without needing large datasets of human gameplay or long calibration periods for online learning.

Adapting to Unknown Conventions in Cooperative Multi-Agent RL

Bidipta Sarkar

bidiptas@stanford.edu

1 INTRODUCTION

When playing a cooperative game, players need to understand the behavior of their teammates for seamless coordination. If two chefs are working together to make a meal, they need to ensure that they are performing a different step in the cooking pipeline from their partner and coordinate on using resources, like the stove or oven. As the team works together for many iterations of the environment, they establish a "convention," which is a solution to the coordination problem that occurs by arbitrarily resolving symmetries the team encounters in the environment Hawkins et al. (2017). Standard multi agent reinforcement learning algorithms mimic the process of convention formation through the process of self-play Gleave et al. (2020), where a team is trained in a joint fashion, often with the same policy network. However, standard self-play is unaware of the existence of other conventions since it is only trained to maximize its reward with its own partner. This means that pure self-play algorithms lack the ability to generalize to different conventions or adapt to a change in partners.

We can measure how well two players with different policies can work together by calculating the expected score when they team up, which is called cross-play. If a particular player has a very high self-play score, but a very low cross-play score with other partners, it has a very narrow understanding of the possible conventions in a game and therefore cannot generalize well to new conventions. An ideal AI agent would instead work decently well with partners of any type by not making assumptions about the partner's convention, but it would also be able to adapt over time once it becomes more confident in the behavior of its partner.

From the perspective of multi-task learning, we can think of the different possibilities of partners as different "tasks" that a specific agent might encounter. We can generate a diverse set of conventions using our CoMeDi algorithm (described in section 4.1), and divide the conventions into a training set and testing set. Using this definition of a task, we can use any single-agent multi-task or meta-RL algorithm to adapt to new agents.

In this project, we wish to construct an AI agent that can adapt to different conventions in a few-shot setting in the game of Overcooked from Carroll et al. (2019). In particular, we generate a common best response to a diverse set of conventions and use few-shot fine tuning to adapt quickly to previously unseen conventions. By creating an adaptive agent, we demonstrate that fine tuning a common best response can help facilitate complex human-AI coordination in domains where humans exhibit a wide variety of behaviors.

2 PRELIMINARIES

We interpret the game of Overcooked as both a multi-agent game and a multi-task single-agent game, where the task is defined by the partner's behavior.

As a cooperative multi-agent game, we model Overcooked as a two player decentralized, partially-observable, Markov Decision Process (Dec-POMDP) (Peshkin et al. (2000)) with identical payouts. The Dec-POMDP $\mathcal{M}$, is the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \mathcal{O}, \gamma, T)$, where $\mathcal{S}$ is the joint state space and $\mathcal{A} = A \times A$ is the joint action space. The transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \{0, 1\}$, is the probability of reaching a state given the current state and joint action. Note that the transition function is deterministic in the case of Overcooked. The reward function $r : \mathcal{S} \to \mathbb{R}$, gives a real value reward for each state transition. The observation

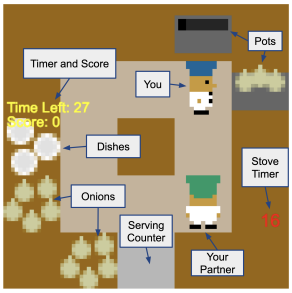

Figure 1: Example of the Overcooked "Coordination Ring" layout.

function, $\mathcal{O} : \mathcal{S} \to O \times O$, generates the player-specific observations from the state. Finally, $\gamma = 0.99$ is the reward discount while $T = 200$ is the horizon.

Each player follows a stochastic policy $\pi^i(a^i|o^i)$, which is the probability that agent i chooses action a^i given its observation o^i. We define a convention as a joint policy of the agents, $\pi = \pi^1 \times \pi^2$, following the definition of a convention as an arbitrary solution to a recurring coordination problem in literature (Lewis (1969); Hawkins et al. (2017)). At time t, the environment is at state $s_t \in \mathcal{S}$, so the agents receive observations $(o_t^1, o_t^2) = O(s_t)$ and sample their actions as $a_t^i \sim \pi^i(a_t^i|o_t^i)$. The environment generates the next timestep as $s_{t+1} \sim \mathcal{P}(s_t, a_t, s_{t+1})$ and the reward as $r(s_{t+1})$. The trajectory is defined as the sequence of states and actions in the environment, which is $\tau = (s_0, a_0, \ldots, s_{T-1}, a_{T-1}, s_T)$. The discounted reward for a trajectory is $R(\tau) = \sum_{t=1}^{T} \gamma^t r(s_t)$. The expected reward for a pair of policies from conventions i and j is $\mathcal{J}(\pi_i, \pi_j) = \mathbb{E}_{\tau \sim (\pi_i^1, \pi_j^2)}[R(\tau)]$.

We also consider the environment *induced* by a fixed partner π_i as a partially-observable, Markov Decision Process (POMDP). This induced environment, $\mathcal{M}_{\pi_i}$, is the tuple $(\mathcal{S}, A, \mathcal{P}_{\pi_i}, r, \mathcal{O}^2, \gamma, T)$. Note that the state space, (decentralized) action space, reward, (decentralized) observation, discount, and horizon are identical to their multi-agent counterparts. The only difference is the transition function, $\mathcal{P}_{\pi_i} : \mathcal{S} \times A \times \mathcal{S} \to [0, 1]$. Since the fixed partner π_i is stochastic, $\mathcal{P}_{\pi_i}(s, a, s') = \sum_{a' \in A} \pi_i(a'|O^1(s))\mathcal{P}(s, (a', a), s')$. In other words, the single-agent POMDP folds the fixed partner into the environment, changing the transition from a deterministic function into a stochastic one. We can also define an expected reward function analogous to the multi-agent case as $\mathcal{J}_{\pi_i}(\pi_j) = \mathbb{E}_{\tau \sim (\pi_i^1, \pi_j^2)}[R(\tau)]$.

2.1 Overcooked Environment

For this work, we study the Overcooked environment Carroll et al. (2019). In particular, we examine the coordination corridor environment as presented in Figure 1. This layout is particularly interesting because there are multiple visibly different conventions. In some conventions, both player continually move in a consistent orientation, either clockwise or counterclockwise. Other conventions utilize the middle square to pass ingredients from one player on the left to the other player on the right.

We use the same observation, action, and reward structure from Carroll et al. (2019). In particular, we use their "Medium Level Planner" which sends observations in the form of distances to objects instead of giving pure visual input. This allows the network to focus on the task of coordinating with partners, which allows multi-level perceptrons with 2 hidden layers to have very expressive behavior.

The blue player is considered the "first player" in the environment, and we define our tasks based on the policy of the blue player. Meanwhile, we want to learn a policy for the green "second player" that can adapt to the blue player.

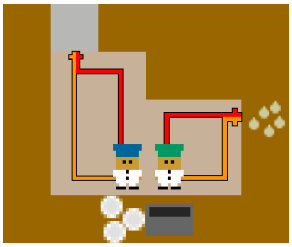

Figure 2: Example of "trivial" variations in conventions. The red convention and orange convention are fully compatible with one another.

When two human players are controlling the two agents, we found that getting a score of 100 (after reward shaping) was possible after coordinating on a strategy. We therefore use a score of 100 as a threshold for human-like performance. Also, each episode takes 40 seconds for a human player to complete, so we would like to have at most a few dozen episodes of fine-tuning.

3 Prior Work

Many recent advances in creating "general" partner agents come from the MARL subfield of zero-shot coordination. One technique, called Off-Belief Learning in Hu et al. (2021), attempts to create a convention-free partner in the game of Hanabi by assuming that self-play actions actually come from a random agent, therefore creating a grounded policy. Another technique, called TrajeDi in Lupu et al. (2021), creates a best response partner to a population of agents that follow different trajectories in the environment. A different paper by Strouse et al. (2021) uses a best response to a manually selected pool of diverse training partners along with snapshots in their training to learn to collaborate with humans in Overcooked. Note that these ZSC papers do not study the domain of few-shot adaptation, so the policies remain fixed across different episodes of interactions.

One related work on conventions in the multi-task setting comes from Shih et al. (2021), which divides a policy into a "task" module and a "convention" module. To train a policy that works with a new partner, they reinitialize the convention module and maintain the task module. However, the process of adaptation requires many episodes of training before converging to a human-like score in their studied tasks.

4 Task Definition

We can frame our problem of adapting to new agents as a multi-task or meta-learning problem. If we have a diverse "set" of conventions, Π, of size n, we can define n different tasks using the induced single-agent environment from each given convention. Specifically, task i is given by the POMDP $\mathcal{M}_{\pi_i}$. We describe the technique for generating diverse conventions and how they will be used for training in the next subsections.

4.1 Diverse Conventions

We use the CoMeDi algorithm (not yet published) to generate our set of conventions. The main idea of CoMeDi is to minimize cross-play as a metric for diversity while ensuring that agents always act in good-faith at all steps using a regularizer called mixed-play.

Although the final goal of this project is to create an agent that has high cross-play with many different conventions, we still want to find policies that have a low cross-play with one another as a training set. If

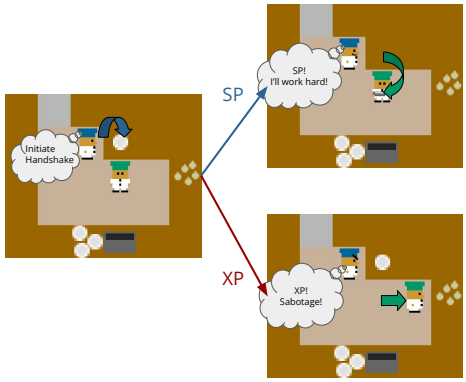

Figure 3: Example of the handshake problem in Overcooked. The blue agent initiates a handshake by placing a dish and chooses its next actions based on the reaction of the partner.

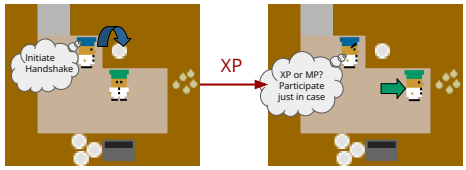

Figure 4: Example of how introducing mixed play (MP) resolves the issue of handshakes.

two conventions have high cross-play with each other, they are similar to one another since a partner for one convention can easily work with the other convention with no changes to its behavior. In figure 2, we see how "trivial" variations can exist between conventions, like slightly different forms of navigation, which do not ultimately affect the other player and therefore result in high cross-play. Statistical diversity techniques like TrajeDi cannot differentiate trivial variations from genuine differences in strategies, because both conventions follow completely separate trajectories. On the other hand, conventions with low cross-play behave differently from one another, so we would like to include both conventions in our diverse set.

Unfortunately, pure cross-play minimization could result in badly behaved policies. In particular, some policies may try to identify their partner as either a self-play partner or cross-play partner. If they are in self-play, they will try to work well in the game, but if they are in cross-play, they might deliberately sabotage the game to artificially minimize the cross-play score, as illustrated in figure 3. We call this issue the "handshake" problem, since players initiate a subtle handshake and gauge the reaction of the partner to determine how it should proceed.

To fix the issue of handshakes, CoMeDi attempts to ensure that policies always act in good faith at every timestep. In particular, a policy should continue acting as if it is working with its own partner even if the past history of actions indicates that it is in cross-play. CoMeDi uses a new technique called "mixed-play" where a state in the environment is randomly generated by sampling between cross-play and self-play until a certain point in time. After this point, the agent continues with pure self-play. Only the pure self-play phase is recorded, and the objective is to maximize the score in this portion of the episode. The agent is never informed of the switch from the random state generation and the self-play evaluation stage, so it must act in good faith at all timesteps in order to not sacrifice the self-play score, as illustrated in 4.

To train a sequence of n conventions, CoMeDi first trains a single agent with the standard multi-agent PPO algorithm Yu et al. (2021). Then, to train convention i, it uses the following loss function:

$$\mathcal{L}(\pi_i) = \mathcal{J}(\pi_i, \pi_i) - \frac{\beta}{2}(\mathcal{J}(\pi_i, \pi^*) + \mathcal{J}(\pi^*, \pi_i)) + \gamma \mathcal{J}_{MP}(\pi_i, \pi^*) \tag{1}$$

where π^* is the convention in the sequence of already discovered conventions that has the highest cross-play with the current convention we are training, and $\mathcal{J}_{MP}$ is the expected score in the self-play phase of mixed-play. Intuitively, CoMeDi finds the convention that is most different from all conventions that have currently been discovered.

4.2 Training and Testing

Using CoMeDi, we train a sequence of 8 conventions using $\beta = 0.5$ and $\gamma = 1.0$ in equation 1. We use the MAPPO algorithm as the base of our implementation, and each policy network is an MLP with 2 hidden layers of size 256. We let our training set be the first 7 conventions and use the final convention as the held-out testing partner. By holding out the last partner, we are essentially finding an adversarial partner to our existing set, which gives us a signal for how well we expect it to extrapolate to different partners.

5 Method

Using the diverse set of training agents generated using CoMeDi from section 4, we can investigate various techniques for adapting to new partners. In particular, we consider the zero-shot technique of constructing a common best response, the few-shot transfer learning setup, and meta-learning with MAML.

5.1 Common Best Response

As the name suggests, the common best response algorithm tries to find a single policy that maximizes the score with all partners in the training set. Using the single-agent POMDP defined in the preliminaries section, we can define the multi-task objective as:

$$\max_{\hat{\pi}} \sum_{i=1}^{n^{tr}} \mathcal{J}_{\pi_i}(\hat{\pi}) \tag{2}$$

where $n^{tr} = 7$ is the number of training tasks. To optimize this loss, we use the Garage implementation of single-agent PPO garage contributors (2019). Our policy and value functions are represented as MLPs with 2 hidden layers of size 64. We train for 7 million timesteps with a batch size of 7000 (corresponding to 35 games per batch) and use an actor and critic learning rate of 10^{-4}. We use the PantheonRL library Sarkar et al. (2022) to transform the environment into a single-agent environment with round-robin partner training.

Since our downstream task is to work with unseen conventions, we do not include a task vector that identifies the partner convention, so we have full parameter sharing.

5.2 Few-Shot Fine Tuning

Using a common best response model, we can fine-tune the weights of the network to adapt to a new partner.

The first technique for fine-tuning directly continues the PPO optimization from the construction of the best response by performing updates at the end of every episode. However, this technique could lead to a policy

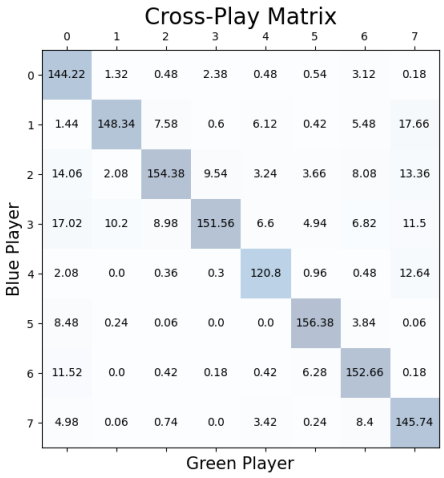

Figure 5: Cross-play score matrix in the Overcooked Coordination Ring layout

that takes bad gradient steps and drifts away from the stable best response policy. The second technique for fine-tuning also updates the model at each episode, but it uses the original common best response model as the starting point for the PPO updates. It uses all the episodes of interactions with the new partner to take PPO updates. In both cases, we fine tune all layers of both the actor and critic networks.

5.3 Meta Learning

We also investigate meta learning as a technique for generating a policy that adapts faster to new conventions. In particular, we experiment with using MAML Finn et al. (2017) using a modified version of the PPO MAML implementation from Garage.

We first try directly training MAML from scratch with a tuned, fixed learning rate. We initialize the policy and value functions with the common best response to see if MAML is able to improve its ability to do one-shot fine-tuning.

6 Results

6.1 Nearest Neighbor Baseline

To ensure that the conventions created using CoMeDi are truly unique, we can construct a cross-play reward matrix to determine if choosing one of the 7 existing test conventions could give a high cross-play score with the held out test convention. In figure 5, we show the cross-play of all pairs of 8 trained agents. Notice how each player has a high self-play score (at least 120), but a low cross-play score with each partner (less than 20). Because of the low cross-play scores, the best score we can expect when the blue player is set to the held-out convention (C7) is less than 5 if we choose any partner from the training set.

These results are exactly what we expect from the CoMeDi algorithm. It is designed to generate a sequence of conventions with a high self-play (very high diagonal in the score matrix), and low cross-play in all other configurations.

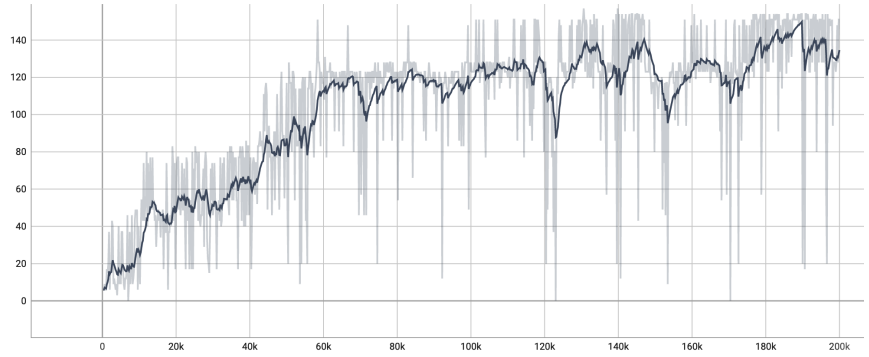

Figure 6: Training a PPO agent from scratch with convention C7. The number of timesteps is on the x-axis and the reward is on the y-axis.

	C0	C1	C2	C3	C4	C5	C6	C7
BR2	86.35	109.81	2.26	39.16	0.15	22.63	6.51	0.09
BR4	112.53	148.46	144.71	123.18	32.45	26.05	30.94	41.54
BR7	107.44	139.41	147.47	110.26	72.67	141.8	81.72	59.58

Figure 7: Scores of Best Responses with CoMeDi partners.

6.2 Training from Scratch

As another baseline, we try to train a PPO agent from scratch with the held-out convention. The "batch size" is equal to one episode (200 timesteps), to mimic the online learning setting. As we see in figure 6, the expected reward becomes 100 after around 300 episodes, which corresponds to over 3 hours of human experimentation data. Based on this result, we need a strong prior model to accelerate training.

6.3 Zero-shot Common Best Response

After generating the sequence of conventions, we can evaluate the technique of forming a common best response. We train 3 different "best response partners" and display the results in figure 7. BR2 is a best response trained against the first two conventions, BR4 is the best response trained against the first 4 conventions, and BR7 is the best response trained against all 7 conventions in the training set. As we can see, adding more training agents aids in extrapolating to the held out agent. We also notice that BR4 typically has higher scores than BR7 among the first 4 conventions, indicating that BR7 has to sacrifice some score for the first 4 conventions to achieve better scores for the others in the test set.

The best responses BR4 and BR7 significantly outperform the nearest-neighbors baseline for the held-out convention. However, the performance is still not at a human level in the zero-shot setting.

6.4 Few-Shot Fine Tuning

Using the generated best response agents, we can use the fine-tuning techniques from section 5 to adapt to partners in a few-shot setting. In particular, we experiment with using less than 20 episodes of online learning, which would correspond to around 13 minutes of human experimentation time. After tuning hyperparameters, we ended up choosing the same hyperparameters from the original PPO run except we make

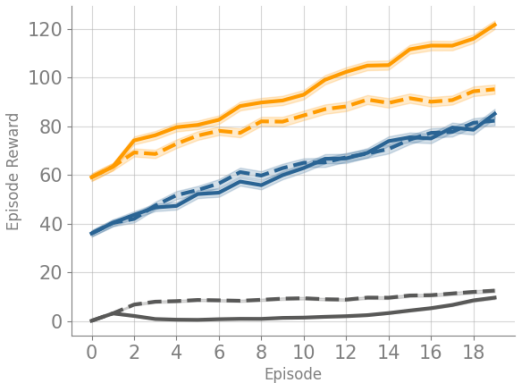

Figure 8: Expected rewards from few-shot fine tuning. Orange starts from BR7, blue starts from BR4, and gray starts from BR2. The dashed lines indicate fine tuning with technique 1 while the solid lines indicate fine tuning with technique 2. The shaded region indicates one standard error of the mean rewards from 500 independent 20-shot trajectories.

the batch size equal to one episode (for real-time fine tuning), and we modified the "ppo epoch" hyperparameter. The "ppo epoch" describes the number of iterations of Adam optimization to take given a batch of trajectories in the environment. We found that increasing ppo epoch from the default of 15 to 50 significantly helped in adapting quickly to new environments.

We present the results of few-shot fine tuning in figure 8. Interestingly, neither technique strictly dominates the other across all pre-trained best response agents. When the pre-trained agent performs poorly at the start, as in BR2, the technique that allows policies to drift (technique 1) actually performs better. However, when the pre-trained agent has a solid prior, like in BR7, the technique that resets the policy back to the start before tuning parameters performs significantly better.

We see that BR7 is able to achieve human-like performance within 12 episodes of fine-tuning with the second technique, which is 8 minutes of human game time.

6.5 Meta Learning

To see if we can gain better one-shot performance, we conducted some experiments using one-shot MAML with a fixed inner learning rate of 0.01 and an outer learning rate of 5×10^{-5} with Adam, which was chosen experimentally by conducting a hyperparameter sweep. The results are presented in figure 9. We see that training MAML from scratch suffers from training instability and results in much lower rewards than the simple common best response model. On the other hand, if we use the common best response model as a prior, the scores improve slightly but its one-shot scores are still around the same level as one-shot fine tuning, resulting in a score of around 70.

7 Conclusion

In this work, we investigate techniques for adapting to new partners in a few-shot setting. We use the CoMeDi algorithm to generate a diverse set of agents, which we use to train a common best response agent. Although the best response agent does not achieve human-like performance when coordinating with a new

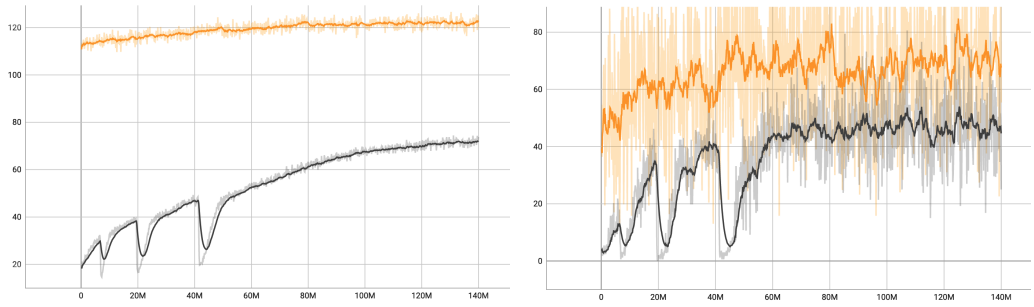

Figure 9: MAML average training reward (left) and average testing reward (right) over 700,000 episodes of training. The black line indicates training from scratch (technique 1) while the orange line indicates training from the BR7 models.

partner in the zero-shot setting, we observe that fine-tuning allows us to achieve human-like performance within a dozen episodes of online play.

The ability to fine-tune in a few-shot setting opens up many possibilities in human-AI collaboration without requiring large datasets of human interactions. We can use the pipeline of generating a diverse set of conventions in a MARL setting, creating a single-agent common best response, and finally fine-tuning to an end user to improve the experiences of human-AI coordination in many cooperative settings.

In the future, we would like to continue improving our fine-tuning algorithm. Although we observed that PPO is able to adapt quickly when boosting the number of ppo-epochs, an algorithm designed for online RL like AWAC Nair et al. (2020) might be able to fine-tune in a more stable manner. We would also like to continue investigating MAML, especially with learned inner learning rates. When we tried training MAML from scratch with a learned inner lr, the training was very unstable, but we might be able to get significantly better performance with a different set of hyperparameters. Finally, we would also like to see how well our technique works in other cooperative games beyond Overcooked.

ACKNOWLEDGMENTS

We would like to thank Andy Shih and Dorsa Sadigh for collaborating on the CoMeDi algorithm.

REFERENCES

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 *Advances in Neural Information Processing Systems*, pp. 5175–5186, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400, 2017. URL `http://arxiv.org/abs/1703.03400`.

The garage contributors. Garage: A toolkit for reproducible reinforcement learning research. `https://github.com/rlworkgroup/garage`, 2019.

Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HJgEMpVFwB`.

Robert XD Hawkins, Mike Frank, and Noah D Goodman. Convention-formation in iterated reference games. In *Proceedings of the 39th Annual Meeting of the Cognitive Science Society*, 2017.

Hengyuan Hu, Adam Lerer, Brandon Cui, Luis Pineda, David J. Wu, Noam Brown, and Jakob N. Foerster. Off-belief learning. *CoRR*, abs/2103.04000, 2021. URL https://arxiv.org/abs/2103.04000.

David Lewis. Convention: A philosophical study. 1969.

Andrei Lupu, Brandon Cui, Hengyuan Hu, and Jakob Foerster. Trajectory diversity for zero-shot coordination. In *International Conference on Machine Learning*, pp. 7204–7213. PMLR, 2021.

Ashvin Nair, Murtaza Dalal, Abhishek Gupta, and Sergey Levine. Accelerating online reinforcement learning with offline datasets. *CoRR*, abs/2006.09359, 2020. URL https://arxiv.org/abs/2006.09359.

Leonid Peshkin, Kee-Eung Kim, Nicolas Meuleau, and Leslie Pack Kaelbling. Learning to cooperate via policy search. In *UAI '00: Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence*, 2000.

Bidipta Sarkar, Aditi Talati, Andy Shih, and Sadigh Dorsa. Pantheonrl: A marl library for dynamic training interactions. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (Demo Track)*, 2022.

Andy Shih, Arjun Sawhney, Jovana Kondic, Stefano Ermon, and Dorsa Sadigh. On the critical role of conventions in adaptive human-{ai} collaboration. In *International Conference on Learning Representations*, 2021.

DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. *Advances in Neural Information Processing Systems*, 34:14502–14515, 2021.

Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.