

Coordination With Humans Via Strategy Matching

Michelle Zhao, Reid Simmons, Henny Admoni

Abstract—Human and robot partners increasingly need to work together to perform tasks as a team. Robots designed for such collaboration must reason about how their task-completion strategies interplay with the behavior and skills of their human team members as they coordinate on achieving joint goals. Our goal in this work is to develop a computational framework for robot adaptation to human partners in human-robot team collaborations. We first present an algorithm for autonomously recognizing available task-completion strategies by observing human-human teams performing a collaborative task. By transforming team actions into low dimensional representations using hidden Markov models, we can identify strategies without prior knowledge. Robot policies are learned on each of the identified strategies to construct a Mixture-of-Experts model that adapts to the task strategies of unseen human partners. We evaluate our model on a collaborative cooking task using an Overcooked simulator. Results of an online user study with 125 participants demonstrate that our framework improves the task performance and collaborative fluency of human-agent teams, as compared to state of the art reinforcement learning methods.

I. INTRODUCTION

Robots increasingly serve as collaborative partners in applications where humans cannot operate alone, such as robot-assisted elder care [1] and cooking [2]. In such applications, humans may employ any number of equally reasonable *strategies* to achieve their goals. Robots designed for collaboration should be able to adapt their behavior in order to coordinate with the different strategies employed by different human partners [3].

Poor coordination between partners can lead to inefficient collaboration on tasks. Consider a robot and human partner working together to make a sandwich. Both partners get condiments, believing the other is getting the bread. This disfluency results in task inefficiencies: production time increases since one agent now must locate the bread before they are able to begin assembling the sandwich; and cleanup time increases since double the amount of condiments have been taken out. In collaborative tasks without explicit verbal communication, teams can be even more susceptible to disfluencies like these. In order to overcome disfluencies, team members must be able to coordinate by inferring their partner’s strategy from their observable actions.

In this work, we explore the following question: *How can a robot partner recognize the task strategy employed by a human partner, and adapt its own response online?* Prior work leverages human-human team demonstrations to learn robot behavior policies for collaborative tasks [4], using

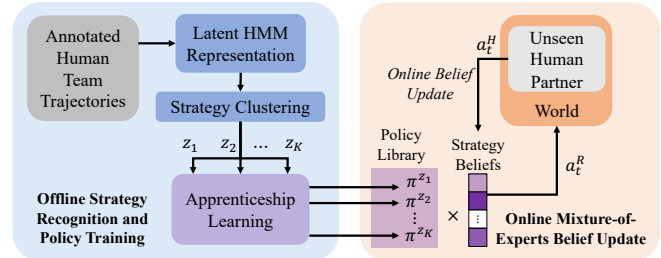


Fig. 1: The offline training paradigm consists of strategy recognition and training a policy library of strategy-specific agent policies. Beliefs over strategies are updated online during interactions with new partners. Actions taken by the robot are sampled from a belief-weighted combination over action distributions generated by each strategy-specific policy.

human data in aggregate, without separating the demonstrations by strategy. When a dataset contains one class that is underrepresented, the trained model often generalizes well to the majority class but poorly to the minority class, a problem addressed by techniques including undersampling [5]. This type of aggregate model performs well with the “average” or most common human behavior, but may miss underrepresented strategies. On the other hand, an agent that can distinguish between strategies will likely generalize better to a greater diversity of human behavior.

We propose a method for a robot to identify and adapt to discrete, task-oriented strategies that determine the team’s behavioral patterns. We situate our work in the simulated Overcooked domain, a human-robot collaborative cooking testbed. Using data collected in [4] of human-human team task demonstrations for five different kitchen environments, we annotate the trajectories to represent them as high level task sequences. Next, we transform the annotated sequences of team actions into low dimensional representations using hidden Markov models. Clustering on the low-dimensional representations extracts groups of similar team behavior, which define discrete strategies representing different team approaches employed on a collaborative task. Robot policies are trained using apprenticeship learning [6] to imitate distinct strategies. The resulting agent is a Mixture-of-Experts model that maintains a dynamic belief over the strategy space for unseen human partners at test time.

We conducted an online user study to investigate the utility of coordination on behavior in human-robot teams. 125 participants performed a collaborative task with our proposed agent as well as with an existing baseline agent [4] in the five environments for which we had data. Our approach improved team task performance in two of the five tested environments and team collaborative fluency in three of the five tested environments.

II. RELATED WORK

1) *Ad-Hoc Teaming*: Ad-hoc teaming in Human Robot Interaction (HRI) requires the ability of robot agents to adapt to unseen partners [7, 8], who may differ in knowledge, skill, and behavior. Prior work [7] proposes a general purpose algorithm that reuses knowledge learned from previous teammates or experts to quickly adapt to new teammates. The approach takes two forms: (1) model-based, which develops a model of previous teammates’ behaviors to predict and plan in online, and (2) policy-based, which learns policies for previous teammates and selects an appropriate policy online. Another important challenge in ad-hoc teaming is modeling uncertainty over partner characteristics [9, 10]. In the Overcooked environment, [4] showed that incorporating human models learned from data improves the performance of agents compared to agents trained to play with themselves. Instead of training agents to partner with a general human proxy model as in [4], we train a library of strategy-specific agent policies that represent different coordination behavior patterns. Distinguishing strategy allows for a policy library that captures differences in team coordination patterns that may otherwise wash out in a single general model.

2) *Multi-agent Reinforcement Learning*: In cooperative multi-agent settings, self-play (SP) trains a team of agents that work well together. A collaborative agent that excels with the partners with which it was trained may not generalize well to new partners at test time, especially when the new partners differ significantly from the pool used for training [11]. Other-play (OP) [12] addresses this problem, demonstrating improved zero-shot coordination with human-AI performance on the Hanabi game [13]. A self-play training paradigm that assembles agents representing untrained, partially trained, and fully trained partners by extracting agent models at different checkpoints in the training duration has been shown to produce robust agents trained on the suite of partners [14]. Prior work [15] models opponents in deep multi-agent reinforcement learning settings by training neural-based models on the hidden state observations of opponents. A Mixture-of-Experts architecture maintains a distribution over different opponent strategies, allowing this model to integrate different strategy patterns.

3) *Adaptation in Human-Robot Interaction*: Past research has studied how robots can adapt and learn from human partners. Key to robot-to-human adaptation is understanding people’s behavior through observation. Markov Decision Processes (MDPs) are a common framework for goal recognition [16]. By learning a model of human intent and preferences [17], robots can reason over different types of human partners [18, 19]. Similar in vein to our work, [20] applied a best-response approach to selecting policies from a library of response policies that best match a particular player type. Building an understanding of the human partner requires multi-faceted models of humans that capture nuanced differences. Our work on adaptation focuses primarily on adapting robot behavior to the task approach (strategy) of a human partner. Our adaptation approach is similar to [21],

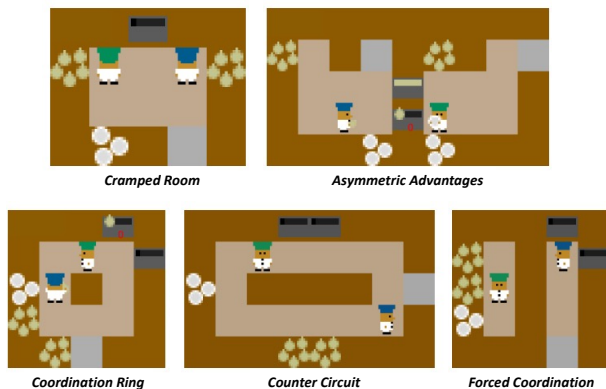


Fig. 2: The Overcooked experimental layouts. Environments vary in the amount of constrained space, actions available to different player positions, and interdependence of player actions to achieve the objective.

where human demonstrations are clustered into dominant types and a reward function is learned for each type, for which Bayesian inference is used to adapt to new users.

III. PRELIMINARIES

1) *Task Scenario*: In order to study human-robot collaboration, we study the *Overcooked* environment [4], a collaborative cooking task. Dyads (consisting of robot agents or humans) collaborate in a constrained shared environment (Fig 2). Their objective is to prepare an order (onion soup) and serve it as many times as possible in an allotted time.

2) *Strategies*: In the Overcooked task, agents must perform sequences of high-level tasks to serve orders. Examples of high-level tasks include picking up onions and plates, placing onions into pots, and serving soup. Each high-level task requires a sequence of lower level subtasks (i.e. motion primitives). Teams collaborate on shared tasks in different ways. For example, in role specialization, players take sole responsibility for particular tasks, whereas in complete-as-needed approaches, each partner performs the next required task. In addition to role-oriented strategies, collaborative approaches also prescribe the order in which tasks are performed. Teams that serve dishes while the next orders are cooking employ more time-efficient strategies. We define collaborative *strategies* as the sequence in which high-level tasks are interleaved and distributed across teammates. Since actions of all team members are involved in task approach, strategy is computed at the team level.

3) *MDP Formulation*: The task is modeled as a two-player Markov decision process (MDP) defined by tuple $\langle \mathcal{S}, \mathcal{A} = \{\mathcal{A}^1, \mathcal{A}^2\}, \mathcal{T}, R \rangle$. \mathcal{S} is the set of states. The action space of a game with two players is $\mathcal{A} = \mathcal{A}^1 \times \mathcal{A}^2$. The set of actions available to each player i is \mathcal{A}^i . The transition function \mathcal{T} determines how the state changes based on a joint action by both players, $\mathcal{T} : \mathcal{S} \times (\mathcal{A}^1, \mathcal{A}^2) \rightarrow \mathcal{S}$. $R : \mathcal{S} \rightarrow \mathbb{R}$ is the team reward function. π^i represents agent i ’s policy. $Z = \{z_1, \dots, z_K\}$ represents the set of possible team collaborative strategies. We further denote a policy that corresponds to strategy z_k as π_k .

IV. APPROACH

We introduce MESH (Matching Emergent Strategies to Humans) as an approach for coordination of collaborative

strategies with human partners. The MESH approach is comprised of two components: strategy recognition and Mixture-of-Experts adaptation (Fig 1). In the strategy recognition step, MESH learns the latent strategies employed by different human teams using unsupervised learning on team trajectories. Since strategies are learned over teams, the MESH agent aims to coordinate on the same strategy employed by the human partner. Once team collaborative strategies are identified, we apply apprenticeship learning to train policies that imitate each extracted strategy. These strategy-specific policies are combined to form a Mixture-of-Experts model [22]. When interacting with unseen human partners at test time, the Mixture-of-Experts model adapts its belief over strategies to the inferred strategy of the human partner.

A. Strategy Recognition

The collaborative strategy of a team is represented as a single learned parameter, which determines how teams order and interweave tasks between members. The robot’s objective is to use observations to identify what strategies exist that human teams may employ. We utilize data collected in [4], containing observed trajectories of human-human dyads in five kitchen environments, illustrated in Figure 2.

We frame identifying collaborative strategies from observations of teams as an unsupervised learning task of recognizing patterns in the sequences of high-level tasks performed by both players. The action space \mathcal{A}^i consists of 5 actions: {Move \times [North, South, West, East], Interact}, where the Interact action encompasses all high-level actions where the agent interacts with the environment by picking up or placing objects. We translate environment interactions into a set of 7 high-level actions (subtasks) $G = \{g_1, \dots, g_m\} = \{\text{Pick-up} \times [\text{Onion, Dish, Soup}], \text{Place} \times [\text{Onion, Dish, Soup}], \text{Serve Soup}\}$. Using these subtask definitions, we construct an annotated dataset $\mathcal{D} = \{\xi_1, \dots, \xi_M\}$ of M team subtask trajectories. The sequence of high-level actions performed by both players defines a team i ’s trajectory: $\xi_i = \{g_1^1, g_1^2, \dots, g_t^1, g_t^2, \dots, g_T^1, g_T^2\}_i$, where g_t^i represents the subtask performed by player i at time t .

Since there are numerous subtask orderings that can define possible team coordination strategies, to maintain a tractable space of strategies to reason over, only orderings that differ significantly should be classified as distinct strategies. MESH learns a low-dimensional sequence representation of team action sequences through a Hidden Markov Model (HMM). HMMs are a formulation for learning probabilistic models of linear sequences that allow a probability for a sequence of observable events to be computed. The HMM is specified by the following components:

- 1) The set of observations to the HMM is $O = \{g_1^1, g_1^2, \dots, g_t^1, g_t^2, \dots, g_T^1, g_T^2\}_{\xi_i \in \mathcal{D}}$, where the sequence of observations is the sequence of high-level tasks performed by both players.
- 2) $X = \{x_1, x_2, \dots, x_N\}$ represents a set of N hidden states.
- 3) $B = X \times X$ is a transition probability from hidden state x_i to hidden state x_j .

- 4) $C = X \times O$ is an emission probability from hidden state x_i to observation g_t^i .

The HMM takes as input the high-level subtask sequences of each team in the dataset. Given these observations, the Baum-Welch algorithm [23], an expectation-maximization approach to learning HMM parameters, is used to learn the transition matrices B and C . We next apply the Viterbi algorithm [24] to compute the most likely hidden state sequence for each sequence of observations, representing an underlying temporal structure in the team action sequences. Since the number of hidden states N is chosen to be less than $|G|$, the hidden state sequence provides a low-dimensional representation, $\bar{x}_i = (x_0, x_1, \dots, x_N)_{\xi_i} \forall \xi_i \in \mathcal{D}$ of each team’s observed sequence of subtasks. The sequences \bar{x}_i are trimmed to length.

The discrete strategy z for each team is learned by clustering over the low-dimensional sequence representation. K-means clustering over the set of hidden-state sequences, $\bar{x}_i \forall \xi_i \in \mathcal{D}$, identifies teams that share the same strategy. The final clusters define strategy classifications $z \in Z$. We performed grid search over the number of strategy clusters K and number of hidden states N , measuring the silhouette score of each combination. Silhouette score [25] evaluates clustering algorithms by comparing how similar an element is to its own cluster (cohesion) compared to other clusters (separation). We selected the combination of number of hidden states and clusters achieving the highest silhouette score to determine the strategy classification for each environment (Figure 3).

B. Mixture-of-Experts Model

1) *Apprenticeship Learning for Training Strategy-specific Agents:* Apprenticeship learning (AL) [26] is the task of learning from expert demonstrations. In order to train agents that employ particular strategies $z \in Z$, we employ the apprenticeship learning algorithm outlined in [26], combined with the Overcooked agent training approach introduced in [4] (Algorithm 1). For each collaborative strategy $z_k \in Z$, we train a human proxy model f_k^h to represent a partner employing strategy z_k . The human proxy model f_k^h is trained using behavioral cloning (BC) on all team trajectories in aggregate with fine-tuning on the subset of trajectories belonging to teams from the same strategy cluster z_k . Next, we learn the reward weights θ_k corresponding to strategy z_k using Maximum Entropy Inverse Reinforcement Learning [27]. The reward features are: $\phi = \{\text{onion placed in empty pot, onion placed in partially-filled pot, dish picked up, soup picked up from pot, both pots full, soup served}\}$. We construct a policy library of strategy-specific agents, trained to employ only their corresponding strategy. Using Proximal Policy Optimization (PPO) [28], we compute the strategy-specific optimal policy π_k for the MDP using rewards $R = \theta_k^T \phi$. Under a similar training regime to [4], the policy π_k is trained with BC partner f_k^h .

2) *Adaptation of Human Partner Strategies:* Leveraging the notion of best response from game theory [29], the adaptation task of the robot at test time is to match its

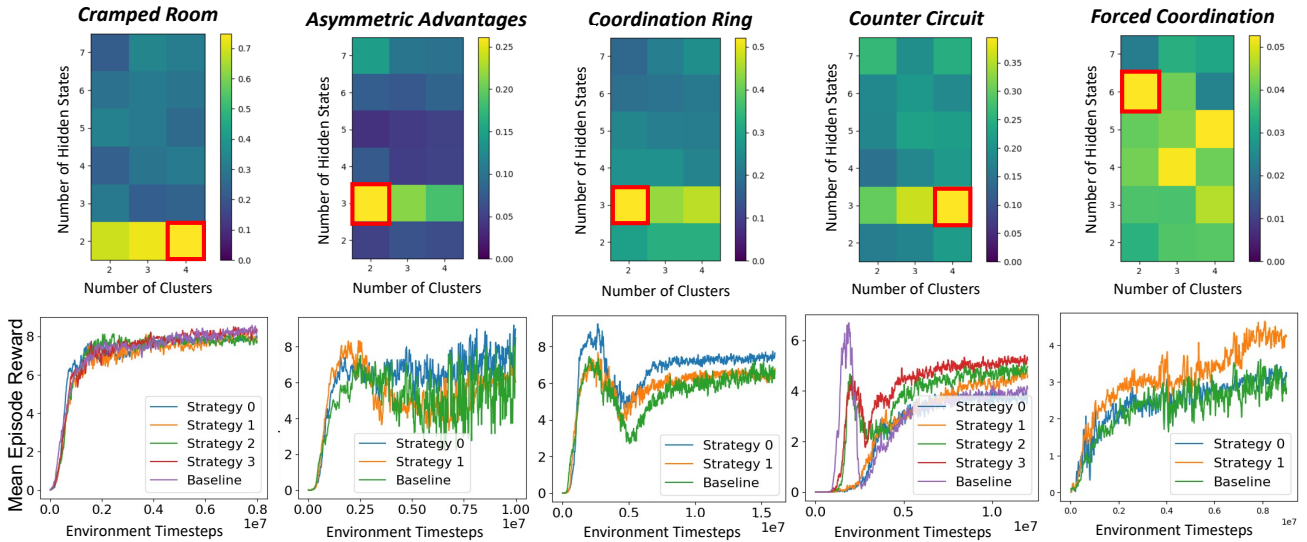


Fig. 3: (Top Row) The strategy clusters for each environment are selected by taking the model achieving the highest silhouette score. Silhouette score is more correlated with the number of hidden states than number of clusters. (Bottom Row) Training rewards for each strategy-specific policy and the baseline. Strategy specific policies achieve varying rewards in training. The same training parameters were used as in the baseline [4] for fair comparison.

Algorithm 1 Apprenticeship Learning of Strategy-Specific Policies

- 1: **for** Strategy $k = 1, \dots, K$ **do**
- 2: $\mathcal{D}_k = \{\xi_i : \text{team } i \in \text{strategy cluster } z_k\}$
- 3: $\theta_k \leftarrow \text{MAXENT IRL}(\mathcal{D}_k)$
- 4: $f_k^h \leftarrow \text{BEHAVIOR CLONING}(\mathcal{D}_k)$ {Train behavior cloning human model}
- 5: $\pi_k \leftarrow \text{PPO}(\theta_k, f_k^h)$ {Compute agent policy using PPO with BC human model partner.}
- 6: **end for**
- 7: STRATEGY-SPECIFIC POLICY LIBRARY = $\{\pi_1, \dots, \pi_K\}$

actions to a new partner’s strategy. The MESH agent uses a discounted-memory Mixture-of-Experts (MOE) model [22] to coordinate with the inferred strategy of the partner. The strategy of the human partner is predicted based on how well each of the candidate strategies in the policy library predict the human’s behavior, with an emphasis on recent human actions. The agent takes a weighted combination over the action probability distributions proposed by each strategy-specific policy. It selects the action with the highest probability after the weighted voting. The next paragraph details the mixture approach (Algorithm 2).

The strategy-specific robot policy library is $\{\pi_1, \dots, \pi_K\}$, where K is the number of strategies identified in the given environment. π_k^R indicates the policy employing strategy z_k at the position of the robot player. π_k^H represents using the strategy-specific robot policy π_k , but taken from the position of the human player. The policies map states to a distribution over actions, which are normalized to form a probability distribution. $\pi_k^R(a_i|s_t)$ denotes a scalar valued probability of action a_i being taken by the robot in state s_t .

The strategy-specific policies are weighted by comparing each policy’s predicted action to the true action taken by the human. When the highest probability action predicted by the strategy-specific policy is the action that the human

Algorithm 2 Online Strategy-Adaptive Mixture-of-Experts

- 1: STRATEGY POLICY LIBRARY = $\{\pi_1, \dots, \pi_K\}$
- 2: strategies = $\{z_1, \dots, z_K\} \in \mathcal{Z}$
- 3: $\mathbf{w}_0 = [\frac{1}{K}, \dots, \frac{1}{K}]$
- 4: **for** $t = 1, \dots, T$ **do**
- 5: $a_t^R = \arg \max_{a_j \in \mathcal{A}} \left[\sum_{k=1}^K \pi_k^R(a_j|s_t) w_t^k \right]$
- 6: Human partner takes action a_t^H
- 7: $d_t^k = \sum_{j=1}^t \gamma^{t-j} \max_{a_i \in \mathcal{A}} \pi_k^H(a_i|s_{t-j}) - \pi_k^H(a_t^H|s_{t-j})$
- 8: $\tilde{d}_t^k = \frac{d_t^k}{\sum_k d_t^k}$
- 9: $w_{t+1}^k = \frac{1 - \tilde{d}_t^k}{\sum_k 1 - \tilde{d}_t^k}$
- 10: **end for**

took, this difference is 0, since $\max_{a_i \in \mathcal{A}} \pi_k^H(a_i|s_{t-j}) = \pi_k^H(a_t^H|s_{t-j})$. Thus, $d_t^k = \sum_{j=1}^t \gamma^j \max_{a_i \in \mathcal{A}} \pi_k^H(a_i|s_{t-j}) - \pi_k^H(a_t^H|s_{t-j})$ is a measure of how inaccurately the robot’s human model represents the human’s actions. The higher d_t^k , the worse the human model. \tilde{d}_t^k is the normalized value of d_t^k over strategies. The belief weights over strategies, \mathbf{w}^t , are the normalized, inverted \tilde{d}_t^k values. The belief model sees only the actions taken by the human partner and does not have access to a true human model. Assuming that the human is sampling from some unknown true human distribution, the expected action taken by the human is maximum likelihood action under their true, but unknown, policy. Because of this limited fidelity feedback from the human, we evaluate the maximum probability predicted human actions of each strategy policy. The γ factor represents a discounted memory factor, which prioritizes recent human actions, allowing the likely strategy of a player to shift as the episode progresses.

V. SIMULATION RESULTS

We first compare the MESH agent to the baseline by evaluating the performance of both agents with a human

Evaluation of MESH vs Baseline with Human (ALL) Proxy Model

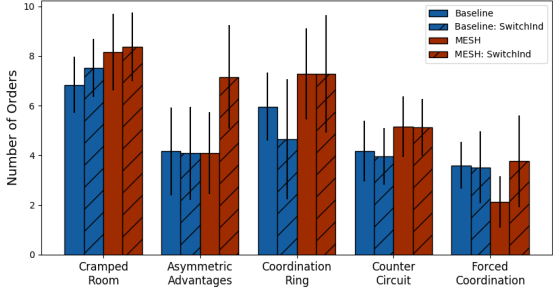


Fig. 4: The MESH agent generally outperforms the Baseline model while collaborating with the human-(ALL) proxy model, except for in one position in the Forced Coordination and Asymmetric Advantage environments. The number of orders is averaged over 25 simulated games. The error bars represent standard deviation.

proxy model trained using behavioral-cloning over all teams in the dataset. We refer this model the human-(ALL) proxy model. Switched indices (SwitchInd) refers to swapping the agent and human player positions. The MESH agent outperforms the Baseline agent in both player positions for four out of five environments. In Forced Coordination, the MESH agent outperforms on one position, while performing worse relative to the Baseline in the other position (Fig 4).

VI. USER STUDY

We ran a user study to compare collaboration of human partners with the MESH agent and a baseline agent that was trained using PPO with a single behaviorally-cloned human partner [4]. Each participant performed a collaborative cooking task with both the MESH agent, and the Baseline agent. The study was a 5×2 within-subjects design. The first repeated variable was (1) Kitchen Environment, of which there were 5 levels (layouts: [Cramped Room, Coordination Ring, Asymmetric Advantages, Counter Circuit, Forced Coordination]). The second repeated manipulated variable was (2) agent adaptability, which consisted of two types: MESH agent (adaptive), and (2) Baseline agent trained with PPO with a behaviorally-cloned human proxy (non-adaptive). The MESH Agent is adaptive since it is trained to imitate different strategies learned from observing human teams, allowing it to adapt its mixture policy to the strategy it believes the human partner is playing.

Participants were given a scaffolded training task to familiarize themselves with the online game’s controls. Next, participants played all 5 environments of Overcooked with each of the two agents, for a total of 10 rounds. Each round is 60 seconds. The 5 environments were played sequentially, meaning that for each environment, participants will play with one type of agent, followed by the other, before moving onto the next environment. The order of environments and the order in which agents were presented was counterbalanced. We compared the two team compositions (human-MESH, human-Baseline) on task performance and collaborative fluency. Collaborative fluency (CF) [30] is defined as the coordination of joint activities by members in a team. After each round, the participant rated the collaborative fluency of the team. Between environments, after playing with both

agents, participants evaluated which agent they preferred.

A. Hypotheses

We hypothesize that participants collaborating with the MESH agent will demonstrate better task performance and higher collaborative fluency, measured through subjective and objective measures.

- H1:** Participants will perform better on the cooking task by serving more dishes when paired with MESH.
- H2:** Participants will prefer to team with MESH agents.
- H3:** Teams with MESH agents will exhibit higher collaborative fluency, measured through objective measures of idle time, concurrent activity, and functional delay.
- H4:** Teams with MESH agents will self-report higher subjective team collaborative fluency.

B. Measures

As we seek to develop a more collaborative agent, collaborative fluency metrics measure how effective an agent is at coordinating its behavior with its human partner. While fluency measures do not always directly correlate with task efficiency, they can change people’s perception of collaboration.

1) **Task Performance:** Task performance is measured by the number of orders served by a team.

2) **Preference:** After each kitchen environment was played, participants were asked to select their preferred partner.

3) **Objective Collaborative Fluency:** Participant trials were analyzed and 3 objective fluency measures were computed (in seconds): (1) human idle time, (2) robot idle time, and (3) concurrent activity [30].

4) **Subjective Collaborative Fluency:** Participants are asked 5-point Likert scale questions rating the team’s fluency, fluency over time, the agent’s ability to understand the actions of the participant, and the predictability of the agent’s actions.

- Partner X and I coordinated our actions well together.
- Partner X and I coordinated our actions better as the episode progressed.
- Partner X perceived accurately what tasks I was trying to accomplish.
- I was able to understand and predict what tasks Partner X was trying to accomplish.

We measured alignment between responses to these questions (Chronbach’s $\alpha = 0.92$ [31]), and the values were added into a single subjective fluency score.

VII. RESULTS

We collected data from 148 Prolific [32] workers over the age of 18. We removed 19 participants who did not complete the study or whose data was saved improperly and 4 participants who performed fewer than 10 keypresses in 1 or more trials (indicating deliberate lack of participation), resulting in 125 total participants. The keypress threshold was selected at 3 standard deviations away from the mean after skew adjustment with a square-root transform. We

analyzed all numerical data using a two-way repeated measures ANOVA followed by post-hoc pairwise analyses with Bonferroni correction for multiple comparisons. Mauchly's test of sphericity was performed on the data for each measure, and Greenhouse-Geisser correction was applied for the ANOVA if the sphericity assumption was not met. In the following graphs, * denotes significance at $\alpha = 0.05$, and ** denotes significance at $\alpha = 0.01$. 58.4% of the participants self-identified as female, 40.0% as male, and 1.6% as transgender, non-binary, or other. Participant ages ranged from 18 to 70 ($M = 27.98$, $SD = 8.61$). The recruitment procedure and study were approved by our Institutional Review Board. The study and hypotheses were pre-registered on Open Science Framework (10.17605/OSF.IO/JFW7T).

1) *Task Performance: H1* is only supported for the Cramped Room and Forced Coordination environments (Fig 5). In these layouts, participants served significantly more orders when paired with the MESH model over the Baseline model. For the other three environments, no significant difference between agents was observed. A two-way repeated measures ANOVA found that was a significant main effect of agent type on score ($F(1, 124) = 23.285, p < 0.001, \eta^2 = 0.158$), where participants scored higher with the MESH agent over the baseline. There was a significant interaction between the layout and agent type, ($F(3.46, 428.51) = 8.275, p < 0.001, \eta^2 = 0.063$). Post hoc testing with Bonferroni correction showed that for Cramped Room ($p = 0.009$) and Forced Coordination ($p < 0.001$), there was a significant increase in number of orders served with the MESH agent.

2) *Preference: H2* is only supported for the Forced Coordination environment. Agent preferences between the MESH and Baseline agents were queried via a forced choice preference question for each environment. A Chi-Squared Test measured the direction of participant preferences between agent types by comparing preferences against a uniform distribution. People preferred MESH for Forced Coordination: ($\chi^2 = 17.67, p < 0.001$). The baseline was preferred for Coordination Ring: ($\chi^2 = 8.712, p = 0.003$). The other environments did not exhibit significant differences.

3) *Objective Collaborative Fluency: Idle Time and Concurrent Activity: H3* is supported in the majority of environ-

Team Performance In Each Environment By Agent Type

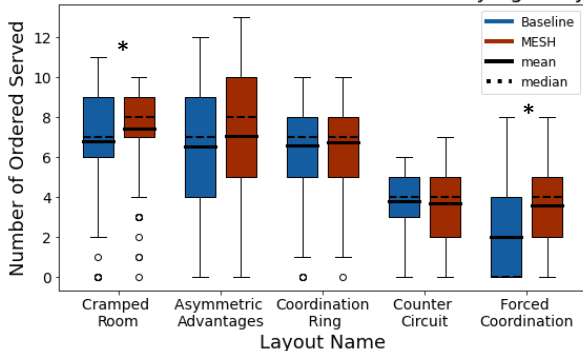


Fig. 5: Scores of human-MESH teams increased in Cramped Room and Forced Coordination environments. No significant difference was found in the other three environments.

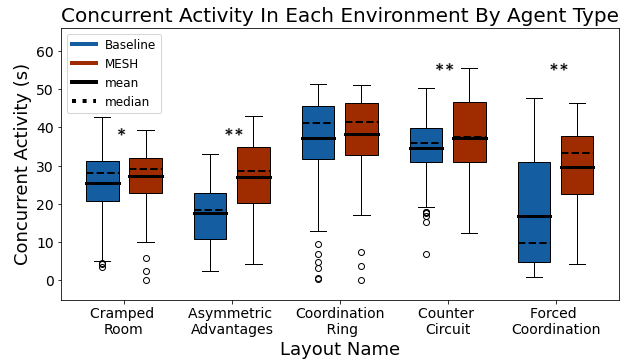


Fig. 6: Concurrent activity between human and robot partner was increased with MESH for four of five environments.

ments in concurrent activity and robot idle time. There was a significant main effect of agent type on the team concurrent activity ($F(1, 124) = 123.846, p < 0.001, \eta^2 = 0.500$), where human-MESH teams displayed higher concurrent activity. There was a significant interaction between the layout and agent type on concurrent activity, ($F(2.611, 323.762) = 27.983, p < 0.001, \eta^2 = 0.184$). Post hoc testing with Bonferroni correction showed that for Cramped Room ($p = 0.018$), Asymmetric Advantages ($p < 0.001$), Counter Circuit ($p = 0.004$), and Forced Coordination ($p < 0.001$), there was a significant increase in concurrent activity with the MESH agent (Fig 6).

The results demonstrated a significant main effect of agent type on the robot idle time ($F(1, 124) = 144.604, p < 0.001, \eta^2 = 0.538$), where human-MESH teams generally experienced lower robot idle time. There was a significant interaction between the layout and agent type on robot idle time, ($F(2.883, 357.535) = 28.523, p < 0.001, \eta^2 = 0.187$). For Cramped Room ($p < 0.001$), Asymmetric Advantages ($p < 0.001$), Coordination Ring ($p = 0.047$), Counter Circuit ($p < 0.001$), and Forced Coordination ($p < 0.001$), robot idle time significantly decreased with the MESH agent (Fig 7).

We found a significant main effect of agent type on the human idle time ($F(1, 124) = 64.700, p < 0.001, \eta^2 = 0.343$), where human-MESH teams experienced lower human idle time generally. There was no significant interaction between the layout and agent type on idle time ($F(2.714, 336.545) = 2.538, p = 0.063, \eta^2 = 0.020$).

Robot Idle Time In Each Environment By Agent Type

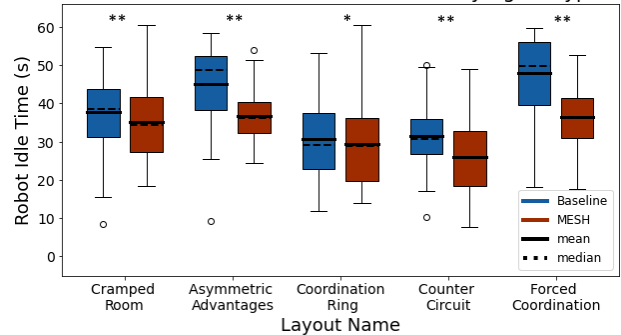


Fig. 7: Participants pairing with MESH agent saw decreased in robot idle time in all environments.

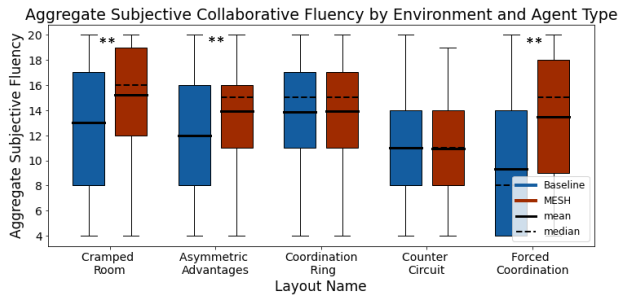


Fig. 8: Subjective collaborative fluency aggregated over the four measures was higher with MESH in three environments, with comparable numbers in the other two environments.

4) *Subjective Collaborative Fluency*: **H4** is supported in the majority of environments. In the Cramped Room, Asymmetric Advantages, and Forced Coordination environment, participants rated the collaborative fluency of the interaction significantly higher when paired with the MESH model over the Baseline model (Fig 8). There was a significant main effect of agent type on the aggregate subjective fluency measure ($F(1, 124) = 23.693, p < 0.001, \eta^2 = 0.160$), and a significant interaction between the layout and agent type ($F(4, 496) = 12.294, p < 0.001, \eta^2 = 0.090$). Post hoc testing with Bonferroni correction showed that for Cramped Room ($p = 0.001$), Asymmetric Advantages ($p = 0.002$) and Forced Coordination ($p < 0.001$), there was a significant increase in self-reported interaction fluency ratings with the MESH agent (Fig 8).

VIII. DISCUSSION

Our results show that an agent designed for coordination on strategy may often positively impact task performance and collaborative fluency of human-robot teams. The benefits to task performance and fluency are exhibited in some, but not all, environments. Human-MESH teams outperformed human-Baseline teams in two environments: Cramped Room and Forced Coordination. The other three environments did not exhibit significant differences in task performance between agents. In the three environments without performance improvement, the MESH algorithm for learning strategy-specific policies likely resulted in behavior similar to the baseline approach. The IRL reward learning under the set of predefined features may not have resulted in significantly different learned rewards for each strategy, causing the agents trained using the learned reward features to display similar behavior to the aggregate human model baseline.

Participants did not show strong preferences for either of the two agents, despite achieving higher score with the MESH agent. This result suggests that the differences between the two agents may have been too subtle to perceive. Agent behavior under a given strategy may not be identifiable from all states, but rather a subset of particular states. The policies trained on learned reward weights may have led to behavior similar across strategies and to the baseline for many states, leading to agent behavior that was not easily distinguishable by participants.

While task efficiency is one measure of agent performance, collaborative fluency can provide insight into the

ability of human-robot teams to coordinate their actions on shared tasks. The interaction with MESH saw higher concurrent activity between the agent and human partner (4 of 5 environments), as well as lower agent idle time (all environments). Participants teaming with the MESH agent also generally experienced lower human idle time; however, since there was no significant interaction between layout and agent type, we did not perform post-hoc analyses on which environments this particularly occurred in. These measures of objective collaborative fluency are related: human and robot idle time influence the amount of concurrent activity. Participants perceived higher collaborative fluency in their interactions with the MESH agent in a majority (3 out of 5) of environments. The other two environments, Coordination Ring and Counter Circuit, which did not report higher subjective fluency, were also among the environments where the MESH agent did not improve task performance. This further suggests that there was not significant difference in agent behavior between strategy-specific policies and the baseline policy. Our results demonstrate that having an agent coordinate on collaborative strategy with a new human partner can offer potential benefits in task performance and collaborative fluency, but the effectiveness varies across environments. The MESH framework for training coordination agents does not underperform compared to the baseline in any environments, suggesting that developing an agent capable of coordinating with different behavioral strategies in human data can assist in better leveraging limited demonstrations.

A key assumption of MESH is that a team that adopts the same strategy will perform better than a team that doesn't. While our study demonstrates that strategy coordination benefits performance and team fluency, there do exist scenarios in which this is not true: it is possible that two different task strategies may be complimentary, and the best response of a robot partner is to employ a complimentary, not identical, approach. We ran a simulated experiment comparing the performance of our strategy-specific policies with each other, and found that there were cases where teams of matching strategies performed worse than non-matching teams, illustrating an example of when direct matching may not be ideal. We additionally assume that the dataset of team observations contains a representative sample of varying team strategies from which we can learn a reasonable library of policies. Under this assumption, by strategically focusing agent training on a subset of the data, we can construct strategy-specific agent policies.

This work represents a step towards developing collaborative agents capable of coordinating their task approaches with human partners. As data gathered about human behavior in different collaboration contexts must increase for better trained agents, this approach provides a candidate unsupervised approach for training agents with an understanding of human strategies and an ability to coordinate with learned, emergent strategies. In future work, we aim to consider alternative reward features, that may more strongly represent differences between strategies, and compare our method to planning-based approaches and approaches that infer the

intentions of the human partner.

IX. CONCLUSION

Coordination of joint actions between humans and robots is required for effective collaboration. We employ a notion of collaborative strategies as approaches teams take towards interweaving team members' actions in performing a shared task. In particular, agents must coordinate their strategies for approaching joint goals in order to facilitate fluent team collaboration. This ability to adapt necessitates the agent first identify existing collaborative strategies and learn policies that employ each strategy. In this work, we develop MESH: a computational framework for learning human collaborative strategies and training an agent with the ability to adapt to the strategies of unseen human partners. An unsupervised clustering-based approach is used for recognizing different teaming patterns from human data. We leverage the learned clusters to train a library of different policies that perform distinct collaborative strategies. We use the identified strategies to construct a Mixture-of-Experts model that adapts to unseen human partners. Results of our user study find that the MESH agent offers higher objective and subjective collaborative fluency over an existing baseline. We see this work as an investigation towards designing robot behavior that reasons about how different task-completion strategies interplay with those of other team members.

ACKNOWLEDGMENT

This research is based upon work supported by the Defense Advanced Research Projects Agency, award number: FP00002636. This work was also supported by NSF IIS-2112633. We thank Timothy Hyun for assistance in data analysis.

REFERENCES

- [1] A. Mertens, U. Reiser, B. Brenken, M. Lütke, M. Hägele, A. Verl, C. Brandl, and C. Schlick, "Assistive robots in eldercare and daily living: Automation of individual services for senior citizens," in *International Conference on Intelligent Robotics and Applications*. Springer, 2011, pp. 542–552.
- [2] H. S. Koppula, A. Jain, and A. Saxena, "Anticipatory planning for human-robot teams," in *Experimental robotics*. Springer, 2016, pp. 453–470.
- [3] R. Aumann and S. Sorin, "Cooperation and bounded recall," *Games and Economic Behavior*, vol. 1, no. 1, pp. 5–39, 1989.
- [4] M. Carroll, R. Shah, M. K. Ho, T. Griffiths, S. Seshia, P. Abbeel, and A. Dragan, "On the utility of learning about humans for human-ai coordination," *Advances in Neural Information Processing Systems*, vol. 32, pp. 5174–5185, 2019.
- [5] A. D. Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *2015 IEEE Symposium Series on Computational Intelligence*, 2015, pp. 159–166.
- [6] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the Twenty-First International Conference on Machine Learning*, ser. ICML '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 1.
- [7] S. Barrett, A. Rosenfeld, S. Kraus, and P. Stone, "Making friends on the fly," *Artif. Intell.*, vol. 242, no. C, p. 132–171, jan 2017. [Online]. Available: <https://doi.org/10.1016/j.artint.2016.10.005>
- [8] A. Dafoe, E. Hughes, Y. Bachrach, T. Collins, K. R. McKee, J. Z. Leibo, K. Larson, and T. Graepel, "Open problems in cooperative ai," 2020.
- [9] N. Agmon, S. Barrett, and P. Stone, "Modeling uncertainty in leading ad hoc teams," in *International Conference on Autonomous Agents and Multi-Agent Systems*, Richland, SC, 2014, p. 397–404.
- [10] P. Stone, G. A. Kaminka, S. Kraus, and J. S. Rosenschein, "Ad hoc autonomous agent teams: collaboration without pre-coordination," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, ser. AAAI'10. AAAI Press, pp. 1504–1509.
- [11] B. Cui, H. Hu, L. Pineda, and J. Foerster, "K-level reasoning for zero-shot coordination in hanabi," *Advances in Neural Information Processing Systems*, vol. 34, 2021.
- [12] H. Hu, A. Lerer, A. Peysakhovich, and J. Foerster, "other-play" for zero-shot coordination," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4399–4410.
- [13] N. Bard, J. N. Foerster, S. Chandar, N. Burch, M. Lanctot, H. F. Song, E. Parisotto, V. Dumoulin, S. Moitra, E. Hughes, I. Dunning, S. Mourad, H. Larochelle, M. G. Bellemare, and M. Bowling, "The hanabi challenge: A new frontier for ai research," *Artificial Intelligence*, vol. 280, p. 103216, 2020.
- [14] D. Strouse, K. McKee, M. Botvinick, E. Hughes, and R. Everett, "Collaborating with humans without human data," *Advances in Neural Information Processing Systems*, vol. 34, 2021.
- [15] H. He, J. Boyd-Graber, K. Kwok, and H. Daumé III, "Opponent modeling in deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1804–1813.
- [16] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *in Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, 2000, pp. 663–670.
- [17] T. Bandyopadhyay, S. Kok, E. Won, D. Frazzoli, W. Hsu, D. Lee, and D. Rus, "Intention-aware motion planning," *Springer Tracts in Advanced Robotics, Algorithmic Foundations of Robotics X*, vol. 86, 02 2013.
- [18] T.-H. Nguyen, D. Hsu, W.-S. Lee, T.-Y. Leong, L. Kaelbling, T. Lozano-Perez, and A. Grant, "Capir: Collaborative action planning with intention recognition," 10 2011.
- [19] S. Nikolaidis and J. A. Shah, "Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy," *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 33–40, 2013.
- [20] H. Li, T. Ni, S. Agrawal, F. Jia, S. Raja, Y. Gui, D. Hughes, M. Lewis, and K. Sycara, "Individualized mutual adaptation in human-agent teams," *IEEE Transactions on Human Machine Systems*, 2021.
- [21] S. Nikolaidis, R. Ramakrishnan, K. Gu, and J. Shah, "Efficient model learning from joint-action demonstrations for human-robot collaborative tasks," in *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2015, pp. 189–196.
- [22] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1177–1193, 2012.
- [23] S. Z. Li and A. Jain, Eds., *Baum-Welch Algorithm*. Boston, MA: Springer US, 2009.
- [24] G. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.
- [25] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.
- [26] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Twenty-first international conference on Machine learning - ICML '04*. ACM Press, p. 1.
- [27] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, ser. AAAI'08. AAAI Press, 2008, p. 1433–1438.
- [28] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.
- [29] D. Fudenberg and J. Tirole, *Game Theory*. Cambridge, MA: MIT Press, 1991, translated into Chinese by Renin University Press, Beijing: China.
- [30] G. Hoffman, "Evaluating fluency in human-robot collaboration," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 3, pp. 209–218, 2019.
- [31] M. Tavakol and R. Dennick, "Making sense of cronbach's alpha," *International Journal of Medical Education*, vol. 2, pp. 53–55, 06 2011.
- [32] S. Palan and C. Schitter, "Prolific.ac—a subject pool for online experiments," *Journal of Behavioral and Experimental Finance*, vol. 17, 12 2017.