A Unconstrained Optimal Policy Pair

In this section, we aim to provide a theoretical explanation of why fully-assistive teachers fail to teach humans in our formulation.

We consider a two-player cooperative Markov Game M with joint action space $A = A^{\mathrm{T}} = A^{\mathrm{S}}$, and the joint action $\mathbf{a} \in A$ is a linear sum of the teacher and the student's action: $\mathbf{a} = a^{\mathrm{T}} + a^{\mathrm{S}}$. In addition, the transition $\mathcal{T}(s'|s, \mathbf{a})$ and reward $R(s, \mathbf{a})$ depends on the joint action $\mathbf{a}$ only. Let the π^{T} and ϕ be the policy for the teacher and student respectively.

The objective function for the teacher-student pair is their total discounted joint reward and is given as

$$J(\pi^{\mathrm{T}}, \phi) = \mathop{\mathbb{E}}_{\substack{a_t^{\mathrm{T}} \sim \pi^{\mathrm{T}}, \\ a_t^{\mathrm{S}} \sim \phi}} \left[\sum_{t=0}^{\infty} \gamma R(s_t, a_t^{\mathrm{T}} + a_t^{\mathrm{S}}) \right]. \tag{1}$$

The optimal policy pair $\langle \pi^{\mathrm{T}*}, \pi^{\mathrm{S}*} \rangle$ is the maximizer of the objective equation in (1) and is given as

$$\langle \pi^{\mathrm{T}*}, \pi^{\mathrm{S}*} \rangle = \mathop{\arg\max}_{\pi^{\mathrm{T}}, \phi} \; J(\pi^{\mathrm{T}}, \phi). \tag{2}$$

This is a common setting in many real-world 2-player cooperative games where the two players share the same action space and their joint action is the combined action from both parties. Moreover, we assume that both agents in this game, namely the teacher and the student, are fully rational, but their knowledge about the game differs: the teacher is omniscient and it knows all optimal policy pairs, while the student knows neither the optimal policies nor the reward function. We are interested in investigating how the two agents with misaligned information would behave in a cooperative game.

Intuitively, the omniscient teacher should aim to transfer the task knowledge to the student through teaching. However, we suspect that without explicitly rewarding the teaching behaviors, if we only use the total discounted reward as the objective function, the teacher-student pair may converge to the optimal pair where the teacher performs all actions and the student does not contribute at all. In this section, we provide a theoretical explanation for this phenomenon.

We first show that within the optimal policy pairs, there exists a pair such that the student does not perform any action and leaves all the work to the teacher. We denote this as the degenerate optimal policy pair.

Remark 1 (Optimal Policy Pair with Degenerate Student Policy) *Consider an MDP of which the transition $T(s'|s, \boldsymbol{a})$ and the joint reward $R(s, \boldsymbol{a})$ only depend on the joint action $\boldsymbol{a} = a^{\mathrm{T}} + a^{\mathrm{S}}$, which is the sum of the 2 players' actions. There exists an optimal pair such that the student does not perform any action. That is, $\exists \langle \pi^{\mathrm{T}*}, \pi^{\mathrm{S}*} \rangle$, such that $\pi^{\mathrm{S}*}(s) = \mathbf{0}$ for all $s \in S$.*

Recall our *Teaching Task*, were given the initial student policy ϕ_0 and the update function U, the student will update its policy ϕ by observing the actions based on the π^{T} and converges to $\pi^{\mathrm{S}*}$ in L iteration. We emphasize that different choices of the target policy $\pi^{\mathrm{S}*}$ have different learning costs borne by the student. For example, a simple $\pi^{\mathrm{S}*}$ can converge in a small number of L, yet it shifts the burden of the task to the partner policy π. On the other hand, a policy $\pi^{\mathrm{S}*}$ comprising a holistic set of skills is difficult to learn, yet the converged student policy can complete the task independently in the end. We argue that though we formulate the task in a cooperative setting, the ultimate goal of teaching is to transfer the skills to the student as much as possible. Therefore, if the performance is measured solely as the joint reward from both players, the teacher may be incentivized to induce a degenerate student policy for its simplicity, which deviates from our true teaching objective.

In this section, we show that without explicitly decomposing the target student policy as an independent set of skills, the teacher may prefer to use the degenerate student policy as the target $\pi^{\mathrm{S}*}$ since it is easier to learn.

Remark 2 *Consider an MDP such that its transition $T(s'|s, \boldsymbol{a})$ and joint reward $R(s, \boldsymbol{a})$ only depends on the joint action $\boldsymbol{a} = a^{\mathrm{T}} + a^{\mathrm{S}}$, which is the sum of the 2 players' actions. The degenerate student policy $\pi^{\mathrm{S}*}(s) = \mathbf{0}, \forall s \in S$ is easiest to teach.*

Proof. The degenerate student policy $\pi^{S*}(s) = \mathbf{0}, \forall s \in S$ is a special case of the skill decomposition under the item response theory, whereas we only have one skill to learn and the skill is not to perform any action. In general, the teaching cost increases with an increasing amount of information the teacher wishes to teach. The degenerate student policy contains the least amount of information and it is task-independent, therefore it incurs the least teaching cost and is the easiest teaching policy. □

Finally, we show that if we only measure the total discounted reward from both parties as the objective, a teaching-cost-minimizing teacher will choose the degenerate student policy. Recall that the transition function and the reward function only depend on the joint actions. In this setting, every optimal policy pair has the same total discounted return. Therefore, if we only consider the objective of maximizing the total discount reward, the teacher is indifferent to the choice of policy pairs within the optimal policy pair set. In addition, since the degenerate policy pair incurs the least teaching cost, a learning-cost-minimizing teacher will prefer teaching the degenerate student policy among all the other policies.

B Experiment Setups

In this section, we describe the experiment setup in detail.

B.1 Overcooked-AI.

Overcooked-AI is a benchmark environment for fully cooperative human-AI task performance and has become a well-established domain for studying coordination. The goal of the game is to cook and deliver as much soup as possible in a limited time. To deliver the soup, agents need to put the 3 ingredients in the pot to cook, pick up a plate, get the cooked soup and deliver the cooked soup to the counter. We decompose the policy into two sub-skills: *putting ingredients in the pot* and *delivering the soup*.

We obtained a diverse set of partners for the whole task and each sub-skill by employing maximum entropy population-based training [1, 2] with PPO [3]. In particular, to train a partially assistive teacher on the sub-skill, we add an auxiliary reward to regularize the teacher's behavior: to train the teacher on the sub-skill *putting ingredients in the pot*, we set a negative reward $r = -100$ when the trained agent put ingredients in the pot; to train the teacher on the sub-skill *delivering the soup*, we set a negative reward $r = -100$ when the trained agent serves the soup. Teachers on both sub-skills were trained with an agent that was allowed to act freely in the task.

The fully-assistive teacher acts optimally and can finish the task independently, the random teacher acts randomly during each interaction, and the student-aware teacher performs one of the two sub-skills based on our proposed teaching strategy. We recruited N=20 (8 females and 12 males) participants and randomly assigned them into groups of three, each with a different teacher. Each human subject was introduced to the rule and objectives of the game and had a trial round before they start to practice. However, they were not exposed to which type of partner they were practicing with.

Each participant was trained for 5 games and then evaluated for 1 game. The first two games were used to initialize the parameters. For the evaluation, we trained a sub-optimal partner to play with the human and count the cumulative reward of the task as the performance measure. During the training, we set $r = \frac{1}{3}$ for putting one ingredient and $r = 1$ for serving one soup. We pre-defined $r = 5$ as the optimal policies' performance to measure student performance. During the evaluation, both agents receive $r = 1$ only when one soup is successfully delivered. We show the efficient strategy in Figure 1. The efficient strategy is 1) put multiple onions on the middle table; 2) to go to the pot; 3) to pick up onions from the middle table; 4) put them on the pot, rather than picking up one onion at a time and put them on the pot. The overall idea is to reduce the number of movements that need to be done to deliver the same amount of ingredients.

B.2 Cooperative Ball Maze.

The Cooperative Ball Maze game requires coordination from both the robot and the human. Each party will hold one side of the maze board and tilt it to move the ball out from one of the two exits. To get the ball out of the maze, both agents need to coordinate fast and accurately. We define two sub-skills *leading the rotation* and *following the rotation*.

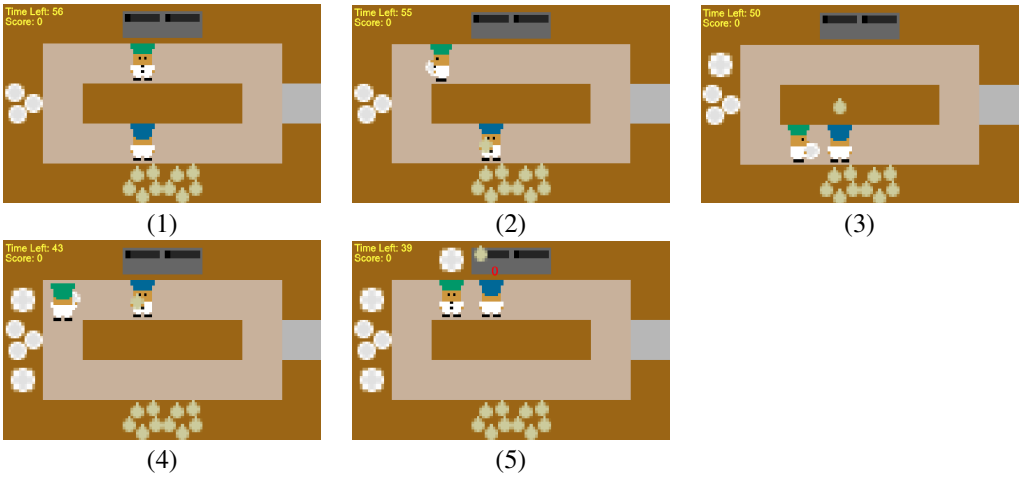

Figure 1. The demonstration of the efficient strategy.

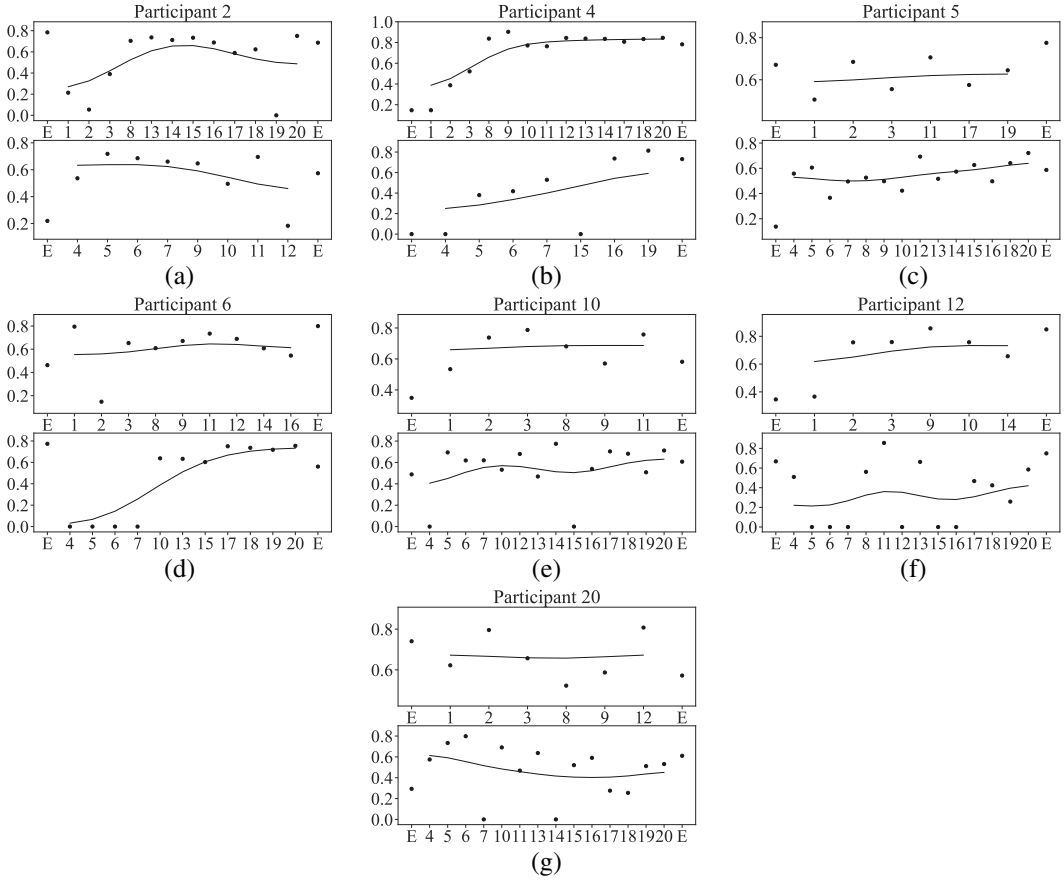

Figure 2. Experiment results of participants of the student-aware teacher's group.

The robot can lead or follow the rotation with a range of compliance. The fully-assistive agent is a maximally compliant agent that always follows the human's lead. The random agent chooses to lead or follow randomly. The student-aware agent starts with a predefined action sequence of 3 robot-following and 3 robot-leading; afterward, it chooses to lead or follow using our proposed decision-making formulation. Both the random and student-aware agents uniformly sample the compliance from a predefined range.

The performance of one interaction is defined as $\max(d_{\max} - d, 0)/d_{\max}$, where d is the duration for the ball to exit, and $d_{\max}$ is an experiment parameter manually defined. In practice, we set

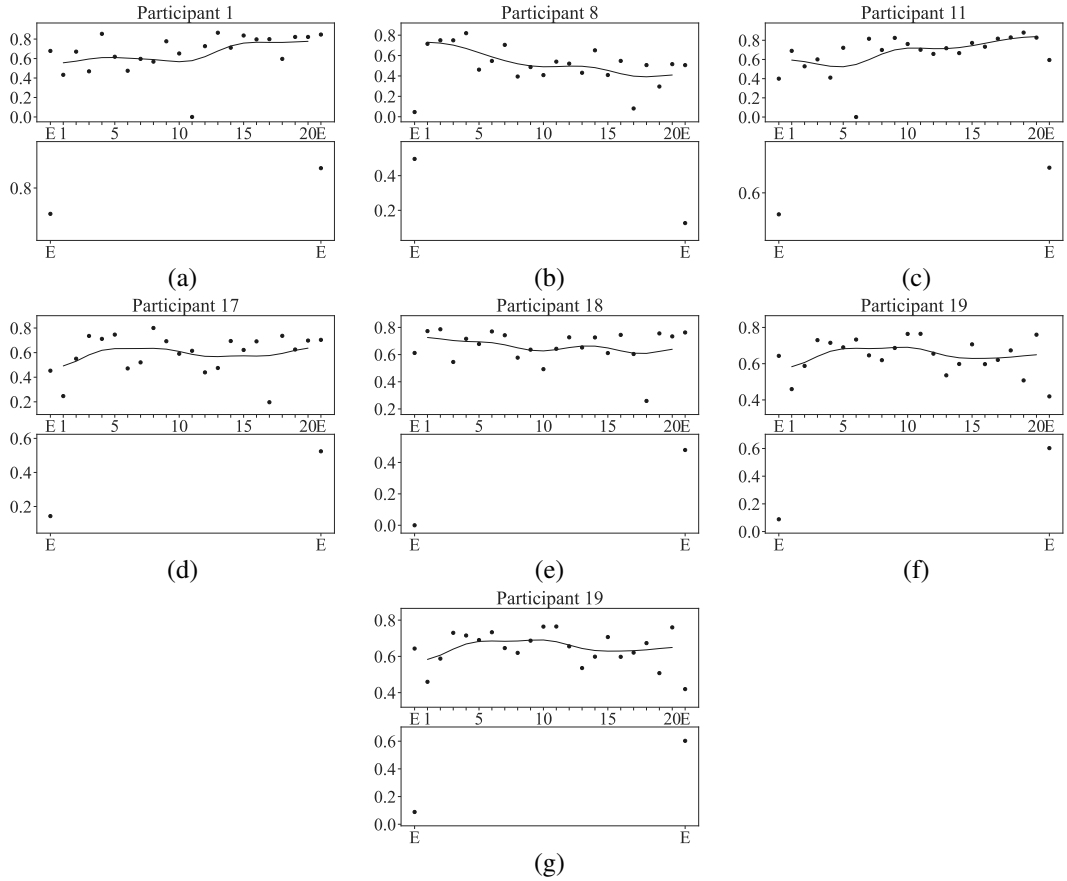

Figure 3. Experiment results of participants of the fully-assistive teacher's group.

$d_{\max}$ to 20 seconds. We measure the overall performance of the participants by averaging the latest scores of the two sub-skills. We recruited N=22 (10 females and 12 males) participants to carry out human-subject experiments. One male participant encountered a hardware issue hence the data was discarded.

The participants were randomly assigned into three groups of fully-assistive, student-aware, and random agents. The participants were first evaluated in the two sub-skills, then trained for 20 interactions, and finally re-evaluated in the two sub-skills. During training and evaluation, the ball is randomly placed in one of the corners on the robot's side. The unseen partner for evaluation has pre-specified compliance and the ball is randomly put on either side of the board. During the 20 interactions, for students in the "random" and "student-aware" groups, the robot's compliance is sampled from a uniform distribution. We set the unseen partner's compliance as the mean of uniform distribution.

We show the raw data points and learning curves of all the participants in Figure 2-4. The top and bottom plots correspond to human *leading* and *following* sub-skills respectively. The vertical axis corresponds to the performance. The horizontal axis corresponds to the indices of the 20 rounds of training interactions. "E" corresponds to the evaluation round. Dots represent the raw data, and solid lines represent the smoothened training data.

C Model Learning

In this section, we show how the parameters λ and α_t, β are learned. We use the student's performance during the interactions to estimate both λ and α_t, β. Let $v_{1:t}$ denote sequences of the student's performance measure against the expert. We have the posterior

$$P(\lambda, \alpha_t, \beta | v_{1:t}) \propto P(v_{1:t} | \lambda, \alpha_t, \beta) P(\lambda, \alpha_t, \beta). \tag{3}$$

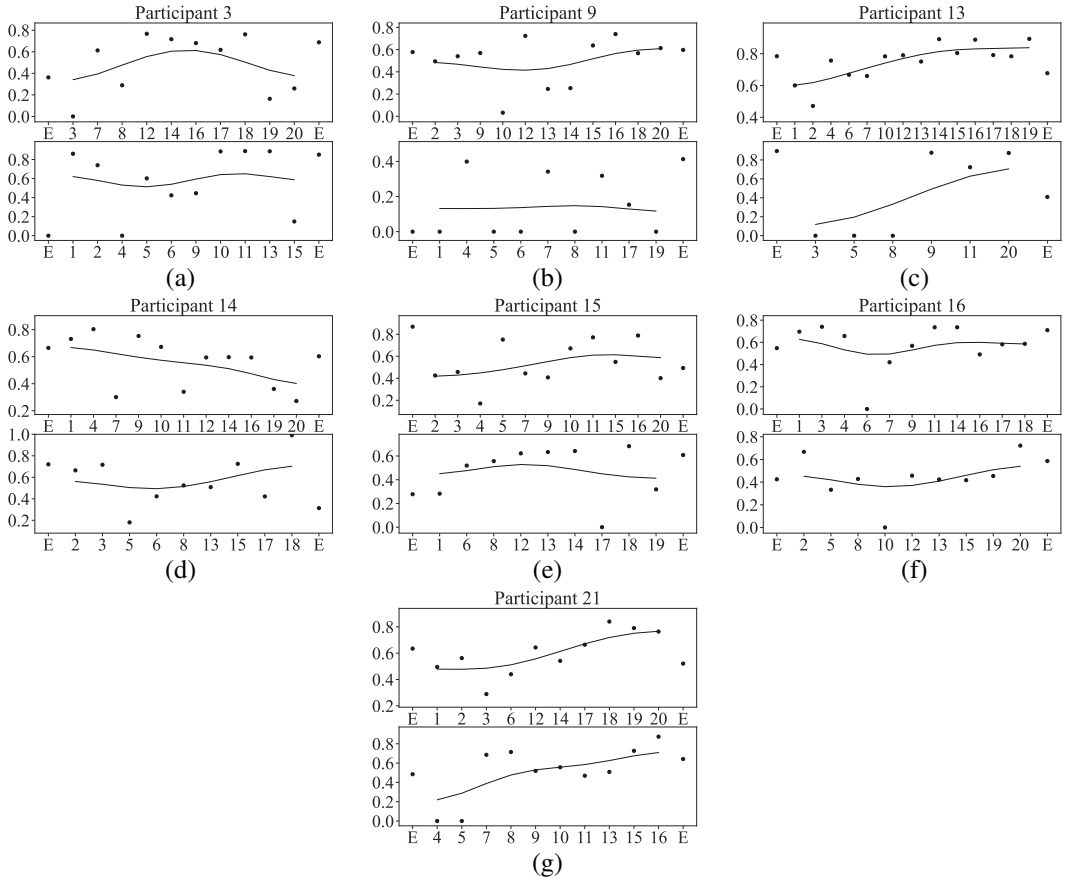

Figure 4. Experiment results of participants of the random teacher's group.

The conditional probability of the observation and current proficiency can be obtained by integrating out all the previous proficiencies. We directly use the results from [4] and assume independence between λ and α_t, β. The approximation of the log posterior over the student's current proficiency given previous responses are:

$$\begin{aligned}\log P(\ \lambda, \alpha_t, \beta | v_{1:t}) \approx \log P(\alpha_t, \beta, \lambda) + \Sigma_{t'=1}^{t} v_{t'} \log \sigma(d_{t'}(\alpha_t - \beta)) \\ +(1 - v_{t'}) \log(1 - \sigma(d_{t'}(\alpha_t - \beta))),\end{aligned} \tag{4}$$

where σ is the sigmoid function, $d_{t'} = (1 + \lambda^{k^2}(t - t'))^{-\frac{1}{2}}$. These parameters for different sub-skills are learned separately. The $d_{t'}$ is often referred to as "effective discrimination" [5], which accounts for discounting the effect of older responses when estimating the current proficiency. Parameters are learned through maximizing a posterior (MAP) with gradient ascent. Empirically we replace the 1PO (one-parameter ogive) model in [5, 4] with 1PL (one parameter logistic) model for computational efficiency since the robot is required to interact with human real-timely. Numerically, they yield similar results.

D Learning From Multi-agent Demonstrations

A multi-agent game is combinatorially more complex than a single-agent game. Collecting the demonstrations for a multi-agent game is much more difficult as we need interactive data among several agents, and finding a suitable model to simulate at least two agents itself is difficult [6]. In addition, to each agent, the environment is non-stationary with distinct optimal demonstrations [7]. The entire state/action space is combinatorially larger than a single-agent game. The success of an LfD policy relies on the coverage and quality of demonstrations. The difficulty in collecting interactive demonstrations, together with the enormously large space of all possible demonstrations, makes direct imitation learning from demonstrations extremely difficult [8].

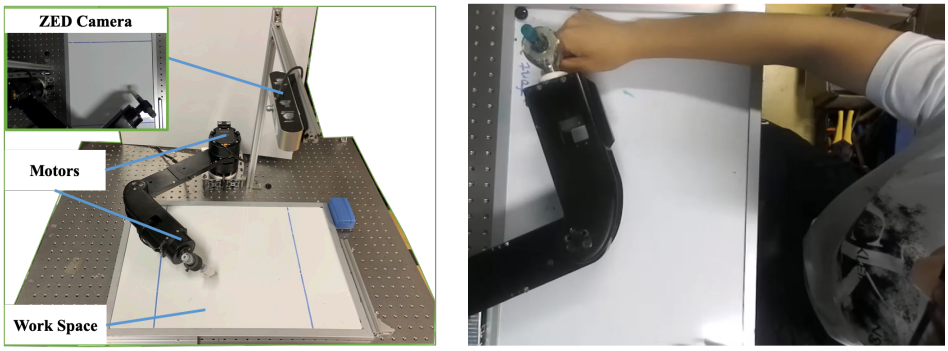

Figure 5. Possible application of cooperative writing.

E Potential Applications

We provide a conceptual framework for cooperative robot teaching and a practical solution to it. Teaching humans to learn to cooperate has a broad field of applications and each of them may require a dedicated solution based on the application. In the following, we provide some potential applications and possible solutions:

1. Training humans in cooperative sports. For example, the framework can be applied to train humans in learning table tennis doubles. The framework can be further extended to train humans in team games with robots as the team members.
2. The framework may also be applied to single-player tasks with proper division of the task. For example, Assisted Chinese calligraphy writing. We show one possible application of Chinese calligraphy writing in figure 5. In this task, the robot would write the Chinese characters together with humans. As there are specified strokes in the Chinese writing system, and they provide natural sub-skill decomposition. Based on a human's performance, the robot could decide on what is the next character to train the human on. In addition, a finer-grained curriculum can be devised to train humans. For example, varying the robot's assistance in terms of force exerted on the human hand to help the human write.

References

[1] A. Lupu, B. Cui, H. Hu, and J. Foerster. Trajectory diversity for zero-shot coordination. In *International Conference on Machine Learning*, 2021.

[2] R. Zhao, J. Song, H. Haifeng, Y. Gao, Y. Wu, Z. Sun, and Y. Wei. Maximum Entropy Population Based Training for Zero-Shot Human-AI Coordination. *CoRR*, 2021.

[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, 2017.

[4] K. H. Wilson, Y. Karklin, B. Han, and C. Ekanadham. Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation. In *Educational Data Mining*, 2016.

[5] C. Ekanadham and Y. Karklin. T-skirt: Online estimation of student proficiency in an adaptive learning system. *Machine Learning for Education Workshop at ICML*, 2017.

[6] P. Hernandez-Leal, M. Kaisers, T. Baarslag, and E. M. de Cote. A survey of learning in multiagent environments: Dealing with non-stationarity. *CoRR*, 2017.

[7] R. Lowe, Y. WU, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, 2017.

[8] J. Song, H. Ren, D. Sadigh, and S. Ermon. Multi-agent generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, 2018.