COACH: Cooperative Robot Teaching

Cunjun Yu[1] **Yiqing Xu**[1] **Linfeng Li**[1] **David Hsu**[1,2]

[1]School of Computing
[2]Smart Systems Insitute
National University of Singapore

Abstract: Knowledge and skills can transfer from human teachers to human students. However, such direct transfer is often not scalable for *physical* tasks, as they require one-to-one interaction, and human teachers are not available in sufficient numbers. Machine learning enables robots to become experts and play the role of teachers to help in this situation. In this work, we formalize *cooperative robot teaching* as a Markov game, consisting of four key elements: the target task, the student model, the teacher model, and the interactive teaching-learning process. Under a moderate assumption, the Markov game reduces to a partially observable Markov decision process, with an efficient approximate solution. We illustrate our approach on two cooperative tasks, one in a simulated video game and one with a real robot.

Keywords: Robot Teaching, Human-Robot Interaction

1 Introduction

How do we teach humans to re-orientate a table jointly or play tennis? Humans often learn by practicing the skills with teachers or partners [1, 2, 3]. This mode of learning is, however, difficult to scale up, as it requires one-to-one interaction and there are not sufficient human teachers [4]. With advances in machine learning, now robots can not only master complex tasks [5, 6, 7], but also collaborate with humans and adapt to human behaviors [8, 9, 10]. In this work, we aim to create *robot teachers* for physical tasks, thus scaling up teaching and providing learning opportunities to many even when human teachers are not available.

Specifically, we propose *Cooperative RObot TeACHing* (COACH), a robot teaching framework to teach humans cooperative skills for two-player physical tasks through interaction. We assume the robot teacher has full knowledge about the task, e.g., a set of optimal policies. The objective is to enable the student to learn the optimal policy as fast as possible. See Fig. 1 for an illustration. COACH treats the teaching task as a two-player Markov game over a *target task*. One player is the robot teacher, and the other is the human student. Under a suitable student learning model, COACH transforms the game into a *partially observable Markov decision process* (POMDP). The POMDP solution enables the robot teacher to adapt to the different behaviors, according to the history of interactions.

One key challenge of COACH is to represent the student's knowledge of the target skills and learning behaviors. First, we leverage *item response theory* (IRT), a well-established framework for educational assessment [11]. IRT provides simplified parametric models that capture the student's knowledge level with respect to the task difficulty in a small number of parameters. COACH treats these parameters as latent variables in the teaching POMDP and learns them from human-robot interaction data by solving the POMDP. Next, to teach complex skills, we draw insights from student-centered learning [12] and human-robot cross-training [13]; we decompose a complex target skill into a set of sub-skills, based on the student's potential roles in the target task. With this compact and decomposed skill representation, we can naturally obtain a partially assistive robot teaching curriculum, in which the human learns the sub-skills one at a time, and the robot assists with the sub-skills not yet learned, to complete the task. While the robot assists the human in the teaching task, its behavior differs from those in common collaborative human-robot interaction settings [14, 15]. There, the primary objective is to complete the task, and the robot is *fully* assistive: if the human does not

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

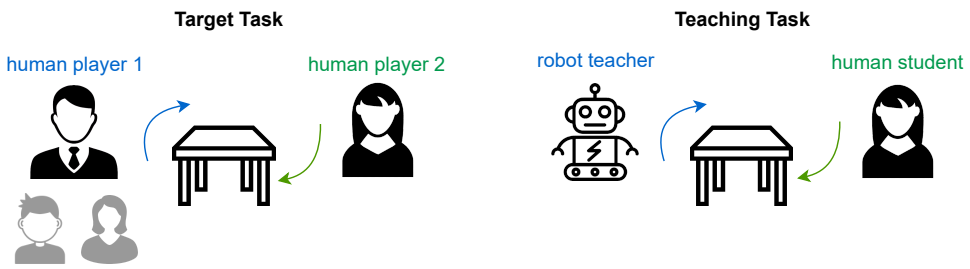


Figure 1. Cooperative robot teaching. In the target task, two human players jointly reorient a table, for example. In the corresponding teaching task, the robot teacher interacts with the human student and teaches cooperative skills so that the student learns to cooperate with partners with varying capabilities or preferences in the target task.

perform, the robot then tries to complete the task on its own, if possible. In the teaching task, the robot is partially assistive and usually avoids assisting with the specific sub-skill to be learned, in order to encourage student exploration and learning.

As a first attempt, we demonstrate an instantiation of solutions to COACH. We conducted human-subject experiments on two challenging human-robot collaboration tasks, Overcooked-AI and Cooperative Ball Maze (shown in Fig. 2), to show the effectiveness of COACH. Our results show that COACH allows us to model and reason over adaptive humans in cooperative teaching. In addition, we find out that a fully-assistive teacher may impede humans' acquisition of skills, and a partially assistive interaction mode indeed motivates students to explore new strategies.

2 Related Work

Assistance in HRI. One major aspect of HRI is how the robot could assist humans with a hidden human objective [14, 15]. The objective of the robot is to infer the human's intention and learns to assist the human. In its simplest form, the action selection and human intention inference are separated [16, 17, 18]. A decision-theoretic framework, assistant POMDP, is developed to capture the general notion of assistance in HRI [19]. The robot integrates the reward learning and control modules to perform sophisticated reasoning over human feedback [20, 21]. However, these two approaches neglect human learning/adaptation and may hinder humans from improving their skills. Our work focuses on how to generate behaviors that facilitate human learning during interactions.

Collaboration in HRI. Another important aspect of HRI is to model interactions as the collaboration between the human and the robot [22], for which the human and the robot share the same objective. However, the joint optimal policy, e.g. rotating the table counter-clockwise, is unknown to both agents in the first place. Their interaction is mutually adaptative [23, 24, 25]. Particularly, as pointed out in [10], if one side is only aware of partial information about the task, the optimal policy pair naturally induces the behavior of active teaching, active learning, and efficient communication between the robot and human. In this work, we focus on the following setting: given that the robot teacher knows the optimal policy, how to carry out active teaching.

Teaching Algorithm for Algorithms. Teaching for algorithms aims to facilitate the learning of the algorithm by choosing or generating training samples. Various teaching techniques including curriculum learning [26] and machine teaching [27, 28, 29, 30, 31] have been effectively applied to supervised learning and semi-supervised learning problems. Similar ideas are further extended to train reinforcement learning agents to learn complex skills, e.g., generate training environment for reinforcement learning [32, 33, 34], choose various demonstrations [35] or learn to decompose the skill [36, 37]. Teaching in cooperative multi-agent RL allows agents to simultaneously become teachers and students for each other [3, 38, 39]. However, such approaches generally require relatively more data for training and to some extent the controlled learning behavior of the learner. Transfer of these approaches to human learning is promising but difficult.

Teaching Algorithm for Human. Despite the aforementioned practical challenges, some algorithms have been successfully deployed for human learning. Attempts on teaching the crowd on classification or concepts prove to be successful [40, 41, 42, 43]. While humans can learn concepts from visual or verbal examples, complex skills like motor control skills can hardly be mastered

through these signals. Recently, skill discovery techniques in reinforcement learning have been introduced to generate a curriculum based on skill decomposition and facilitate humans to learn motor control skills [44]. It focuses on how to adaptively decompose the skill into learnable sub-skills for a human to practice on its own and achieves promising results. Here, we seek to automate the teaching process for humans to cooperate in a physical task and provide a framework for this teaching mode, e.g., table co-reorientation.

3 Cooperative Robot Teaching

We formalize Cooperative Robot Teaching by identifying four key elements: (1) the target task, (2) the student learning model, (3) the teacher model, and (4) the interactive teaching-learning process.

Target task. In this paper, we focus on teaching in *a two-player cooperative task* and we call it the *Target Task*, which is the original cooperative game both agents aim to solve.

Definition 1 . *The target task is a two-player cooperative Markov game between two agents,* 1 *and* 2*, that can be described by a tuple,* $\mathcal{M} = \langle S, \{A^1, A^2\}, T, R, \gamma \rangle$ *with the following definitions:*

- S *a set of target task states:* $s \in S$.
- A^1 *a set of actions for the agent* 1*:* $a^1 \in A^1$.
- A^2 *a set of actions for the agent* 2*:* $a^2 \in A^2$.
- T *a conditional distribution on the next target task state, given the previous state and the actions of both agents:* $T(s'|s, a^1, a^2)$.
- R *a target task reward function that maps target task states and players' actions to real numbers.* $R : S \times A^1 \times A^2 \to \mathbb{R}$.
- γ *the discount factor in the target task.*

At each step, 1 *and* 2 *both observe the current task state* s_t*, then select their actions* $a^1_t \sim \pi^1$ *and* $a^2_t \sim \pi^2$ *respectively.* π^1 *and* π^2 *refers to the policy of agent* 1 *and* 2*. They will receive a joint reward* $r_t = R(s_t, a^1_t, a^2_t)$*. The next state is updated as* $s_{t+1} \sim P_T(s_{t+1}|s_t, a^1_t, a^2_t)$*.*

Given the definition of the target task, we first answer how to represent the knowledge/skills. In this work, we choose to represent a *skill* by the optimal policy π^* to the target task. The optimal policy maximizes the expected cumulative reward when the student is cooperating with a given partner with policy π. Without loss of generality, we assume the ego-agent is agent 2. We define the optimal policy as $\pi^* = \arg\max_{\pi^2 \in \Pi} \mathbb{E}_{\substack{a^1_t \sim \pi, \\ a^2_t \sim \pi^2}} [\sum_{t=0}^{\infty} \gamma^t R(s_t, a^1_t, a^2_t)]$. For example, in the table co-reorientation task, the agent needs to learn to deal with either stubborn or adaptive partners. We recognize that there are other ways to represent knowledge/skills, such as a set of demonstrations and the ground-truth reward function. However, such representations are indirectly linked with the skill's performance; therefore, evaluating its proficiency is more obscured. We choose the optimal policy as the representation since it can be directly optimized over and evaluated.

Student. The student policy is non-stationary since it will improve along with teaching. We model this evolutionary behavior with a tuple of initial policy and an updating function $\langle \pi^S_0, U \rangle$. The student policy represents the student knowledge state. The updating function U models how the student changes its policy after each teaching step.

Teacher. We define the teacher as a knowledgeable agent (expert) who knows a set of optimal policies Π^* for a target task. The teacher aims to acquire a teaching policy π^Tthat can teach any $\pi^*_i \in \Pi^*$ to the student effectively. In this general setting, the choice of the optimal student policy π^*_i depends on the capability, preference, and current knowledge level of the student. A principled approach to selecting the optimal student policy needs to consider the student's preference, his/her update model for the knowledge level, and an estimate of his/her current capability, which requires insight beyond our work. In this paper, we assume that we have an oracle to choose the optimal student policy $\pi^* \in \Pi^*$ to teach, such that this policy π^* matches the preference of the student. The teacher can be described by a tuple of a student's optimal policy and the corresponding teaching policy, $\langle \pi^*, \pi^T \rangle$.

Interactive teaching-learning. We now refer to the robot as the teacher and the human as the student. In the target task, the teacher knows the task's optimal policy π^* while the student does not. The teacher's goal is to act in the most informative way so that the student learns π^* fastest. The

choice of π^* should account for the student's preferences. To embed the objective of teaching and distinguish it from the *Target Task*, we define it as the *Teaching Task* in the following way:

Definition 2 . *The teaching task is a POMDP $\mathcal{M}'$ for the teacher given the target task $\mathcal{M} = \langle S, \{A^1, A^2\}, T, R, \gamma \rangle$, the student $\langle \pi_0^S, U \rangle$, and the optimal policy to teach π^*. It can be described by a tuple $\mathcal{M}' = \langle \bar{S}, \bar{A}, \bar{T}, \bar{O}, \bar{Z}, \bar{R}, \bar{\gamma} \rangle$ with the following definitions:*

- $\bar{S}$ *a set of teaching task state: for $\bar{s} \in \bar{S}$, $\bar{s} = \{s, \pi^S\}$.*
- $\bar{A}$ *a set of actions: $\bar{A} = A^1 \times A^2$.*
- $\bar{T}$ *a conditional distribution on the next teaching task state: $\bar{T}(\bar{s}'|\bar{s}, \bar{a})$.*
- $\bar{O}$ *a set of observations: for $\bar{o} \in \bar{O}$, $\bar{o} = \{a^S, s, r\}$.*
- $\bar{Z}$ *the observation model: $\bar{Z}(\bar{o}|\bar{a}, \bar{s})$.*
- $\bar{R}$ *the teaching task reward function that measures the effectiveness of teaching. $\bar{R} : \bar{S} \times \bar{A} \times \bar{S} \to \mathbb{R}$.*
- $\bar{\gamma}$ *the discount factor in the teaching task.*

The objective of the teaching task is to derive a teaching policy π^T, enabling students to learn π^* for the target task fastest. More specifically, the teacher can influence the student through interactive actions $\bar{a} \in \bar{A}$. Given the student initial policy π_0^S and the update function U, the goal of the Teaching Task is to find a teaching policy π^T that allows $\pi_0^S \to \pi^*$ as fast as possible.

Next, we introduce our choice of the teaching policy π^T, the update function U, and the reward function $\bar{R}$. To devise a student-aware teaching strategy, apart from the current state s_t and the target policy π^*, our π^T also takes the history of observation as input. The history of observation is $h_t = [(s_0, r_0, \bar{a}_0, a_0^S), ..., (s_t, r_t, \bar{a}_t, a_t^S)]$. The action of the teacher then can be sampled from the policy, i.e., $\bar{a}_t \sim \pi^T(\bar{a}_t|h_{t-1}, s_t, \pi^*)$. The student updates π^S with any arbitrary iterative functions conditioned on the history of interactive actions and the joint rewards in the following form: $\pi_{t+1}^S = U(\pi_t^S, h_t)$. Moreover, to incentivize the teacher to speed up the teaching process, we introduce a step-wise teaching cost to the teacher $c_t = C(s_t, \bar{a}_t)$ to penalize unnecessary teaching actions. To this end, we define the reward function as

$$\bar{R}(\bar{s}, \bar{a}_t, \bar{s}'; D, C, \pi^*, \omega) = D(\pi_t^S, \pi^*) - D(\pi_{t+1}^S, \pi^*) - \omega C(s_t, \bar{a}_t), \tag{1}$$

where $\bar{s} = \{s_t, \pi_t^S\}$, $\bar{s}' = \{s_{t+1}, \pi_{t+1}^S\}$, ω is the weighted factor to trade-off the teaching cost and teaching efficiency, and D can be an arbitrary distance measure between two policies, e.g., initial state value in the target task. The solution to the POMDP $\mathcal{M}'$ is a teaching policy π^T that maximizes the expected sum of rewards $\mathbb{E}_{\bar{a}_t \sim \pi^T}[\sum_{t=0}^{\infty} \bar{\gamma}^t \bar{R}(\bar{s}, \bar{a}_t, \bar{s}')]$.

4 Method

In this section, we provide a solution that grounds all the ingredients in the conceptual framework of COACH. The main spirit of our solution is to parameterize students' knowledge state with IRT and decompose complex tasks into a set of role-based independent skills. This enables easier evaluation of the students' proficiencies and provides a ground to derive the partially assistive interaction mode. To begin with, we first define the action space, $\bar{A}$.

Algorithm 1 Approximated Solution to the Teaching Task

Require: Maximum Interactions L, Predefined Interactions N

for $k \in \bar{A}$ **do**:
  Randomly initialize λ and α_t, β, and $\mathbb{X} = \{\}$
  for $i = 1, 2, ..., N$ **do**:
    $\mathbb{X}$.add(v_i)
  end for
end for
for $i = 1, 2, ..., L$ **do**:
  for $k \in \bar{A}$ **do**:
    Learn λ and α_t, β from $\mathbb{X}$
  end for
  $k \leftarrow$ Action selection from λ and α_t, β
  $v_i \leftarrow$ Performance measure from interactions
  $\mathbb{X}$.add(v_i)
end for

4.1 Actions

The action space is constructed through sub-skill decomposition. Sub-skills decomposition is well-studied for single-agent tasks [45, 46, 47]. However, extending the same idea to a multi-agent setting is still challenging since task completion relies on the interaction among multiple parties. We observe that in a multi-agent game, the task naturally comprises several roles, of which each agent takes a subset. The well-established leader-follower model is a particular choice of role-based skill decomposition [48, 49, 50, 51]. Therefore in our work, we propose to decompose skills based

on role allocation. We divide the skill into K locally-independent teachable sub-skills according to the student's potential roles in the task. The teacher's action space $\bar{A} = \{k : k \in \mathbb{Z},\ 0 \leq k < K\}$ consists of teaching each sub-skill. Such a decomposition of skills naturally leads to a partially assistive mode of interaction.

4.2 States

The state space is constructed with Item Response Theory (IRT) [11]. IRT provides a parametric form to represent student's skill levels. Given the limited interactions, we adopted the simplest form, the one-parameter logistic model (1PL), to model human skills. In the 1PL model, each sub-skill $k \in \bar{A}$ is assigned a parameter β^k representing the difficulty, and a parameter α^k called the *proficiency* representing a student's knowledge state. The probability that a student has mastered sub-skill k is given by $P(k) := \sigma(\alpha^k - \beta^k)$, where σ is the sigmoid function. Hence, instead of representing the state with the student's policy π^S, we use $(\alpha, \beta)^K$ to represent the hidden state. That is, for $\bar{s} \in \bar{S}, \bar{s} = \{s, (\alpha, \beta)^K\}$, where $(\alpha, \beta)^K$ is hidden. For each student and each $k \in \bar{A}$, we assume that α changes over time while β does not.

4.3 Transitions

The transition model consists of two main parts, the target task transition model T, and student's update function U. While the former one is known to the teacher, we need to make assumptions on the latter one. Since we define the state space over student's proficiency α in Sec 4.2, the transition model is also constructed over the proficiency. Following the previous work on online estimation of student proficiency [52, 53], for each sub-skill, we model the student's proficiencies over time as a Wiener process: $U(\alpha_{t+\Delta t}|\alpha_t) = \exp\left(-\frac{(\alpha_{t+\Delta t}-\alpha_t)^2}{2\lambda\Delta t}\right)$, where Δt refers to the step interval and λ is a parameter controlling the "smoothness" with which student's proficiency varies over time. For each student and for each $k \in \bar{A}$, we assume λ does not change over time and is learned for each sub-skill respectively. To this end, we construct the transition model in the POMDP as $\bar{T} = \{T, \bar{U}\}$, where T is the transition function in the target task.

4.4 Observations

The observation is composed of the target task state, the student's action and the reward received, $\{a^S, s, r\}$. Recall that in Sec 4.1, we define the action as choosing one sub-skill to train the student, which is a macro-action. For teaching sub-skill k, we redefine the observation as the ratio between the target task rewards achieved by the student's current and optimal policies: $v := \frac{R(s,\bar{a},a^S)}{R(s,\bar{a},a^*)}$, where a^* is the action generated by the optimal policy π^* given the same the target task state s. Since all the sub-skills are treated equally, we will omit the index k for simplicity in the following discussion. Therefore, for $\bar{o} \in \bar{O}$, $\bar{o} = \{s, v\}$. Unlike the binary response in conventional knowledge tracing, the response f we have is continuous and we assume the teacher will only teach one sub-skill at a time. Thus, we use the continuous Bernoulli distribution to construct the observation model: $Z(v|P(k)) = P(k)^v(1 - P(k))^{1-v}$, where k is the sub-skill being taught when v is observed. As a result, the observation model can be defined as $\bar{Z} = \{I, Z\}$, where I is an identity mapping for the observable target task state, $I(s) = s$.

4.5 Reward

The distance between the student's policy and the optimal policies can be represented using $P(k)$. We represent the distance as the average of one minus master probabilities of each sub-skills $D(\pi^S, \pi^*) = \frac{\sum_{k=0}^{K} 1-P(k)}{K}$. There are other ways to specify the goal according to the decomposition of the skill, e.g. weakest or multiply [54]. We choose the sum due to our local-independence assumption on sub-skills. In this work, we assume the cost is uniform, thus, given a finite horizon of interactions, maximizing the reward function defined in Equation (1) is equivalent to maximizing $\bar{R}(\bar{s}, \bar{a}_t, \bar{s}') = \frac{\sum_{k=0}^{K} P_{t+1}(k)-P_t(k)}{K}$, where $P_t(k) = \sigma(\alpha_t^k - \beta^k)$.

4.6 Model Learning and Decision Making

We use the student's performance during the interactions to estimate both λ and α_t, β. Parameters for each sub-skill are learned separately, thus, we omit k for simplicity. Let $v_{1:t}$ denote sequences of student's performance measure against the optimal policy. We have the posterior $P(\lambda, \alpha_t, \beta | v_{1:t}) \propto P(v_{1:t} | \lambda, \alpha_t, \beta) P(\lambda, \alpha_t, \beta)$. The conditional probability of the observation and current proficiency can be obtained by integrating out all the previous proficiencies. The likelihood can be approximated through $P(v_{1:t} | \lambda, \alpha_t, \beta) \approx \prod_{t'=1}^{t} \int P(v_{t'} | \lambda, \alpha_{t'}, \beta) U(\alpha_{t'} | \alpha_t) \mathrm{d}\alpha_{t'}$. An approximation of the log posterior over the student's current proficiency given previous responses can be derived to learn the parameters λ and α_t, β. Following [52, 53], we employ maximum a posteriori estimation (MAP) to learn these parameters. Given the estimation of current state using the past history, we use one-step look-ahead to reduce the impact of the inaccuracy in the transition function. At timestep t, the teacher's action is given as

$$\bar{a}_{t+1} = \arg\max_{k \in \bar{A}} \int U(\alpha_{t+1}^k | \alpha_t^k) P_{t+1}(k) \, \mathrm{d}\alpha_{t+1}^k - P_t(k). \tag{2}$$

In practice, the student is asked to perform on each sub-skill for a few interactions to initialize the parameters.

4.7 Training on Sub-skills

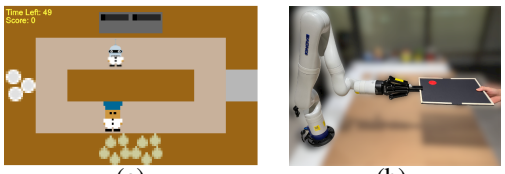
Our overall strategy for training students on each sub-skill is to diversify scenarios the student would encounter during training. Training students on sub-skills naturally leads to a partially assistive partner on unlearned sub-skills, which allows the student to explore the sub-skill freely. We adopt an intuitive assumption: *an agent learns cooperation better with a diverse group of partners*. Such a teaching strategy is effective when dealing with synthetic students [55, 56]. The student could learn from a diverse set of partially assistive partners or learn to cope with them by acquiring new skills.

5 Experiments

We carried out two human-subject experiments to demonstrate how COACH works, one in simulation (Overcooked-AI [57]) and the other with a real robot (Cooperative Ball Maze). Experiment setups are shown in Figure 2. We investigated the teaching performances of three types of teachers: the **fully-assistive** teacher who performs optimally concerning the student's initial capability, the **student-aware** teacher who behaves according to our teaching strategy, and the **random** teacher. The random teacher in the Cooperative Ball Maze experiment chooses sub-skills randomly, while the random teacher in the Overcooked-AI experiment executes actions randomly.

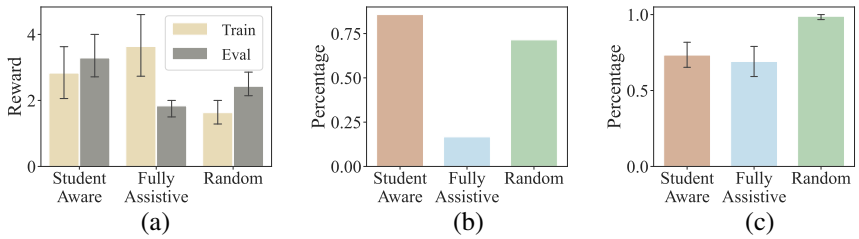
5.1 Setups

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 [58, 59, 60, 61]. The goal of the game is to cook and deliver as much soup as possible in a limited time. We decompose the policy into two sub-skills: *putting ingredients in the pot* and *delivering the soup*. To put ingredients in the pot, there exists one *efficient strategy* to pass the ingredient through the middle table. In brief, rather than picking up one onion at a time and putting them into the pot, the efficient strategy is 1) put multiple onions on the middle table; 2) go to the pot; 3) pick up onions from the middle table; 4) put them into the pot. The overall idea is to reduce the number of movements needed to deliver the same amount of ingredients. We recruited N=20 (8 females and 12 males) participants and randomly assigned them into three groups, each with a different teaching strategy. Students are trained with different teachers and are evaluated with a common unseen partner. We emulate the human partner



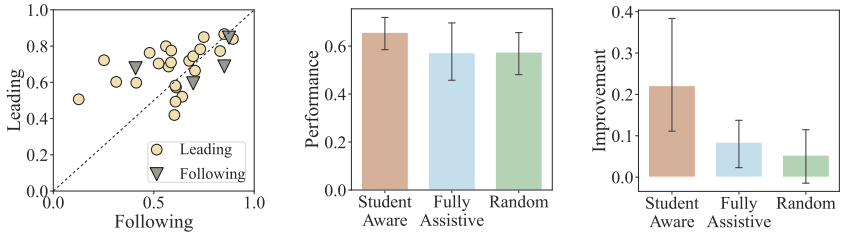
(a) (b)

Figure 2. Experiment setups. (a) Overcooked-AI layout: human participants control the "chef" and the robot controls the "robot". (b) The real robot setup of Cooperative Ball Maze with a simplified setting.



(a) (b) (c)

Figure 3. Results of the Overcooked-AI experiment. (a) Rewards achieved together by the human-robot pairs during training and evaluation. The error bars correspond to the 95% confidence intervals (95%CI). The student-aware teacher outperformed the fully assistive and the random teachers in terms of the evaluation reward (with one-sided p-values 0.001 and 0.03). (b) Percentage of students who found the efficient strategy. None of the students are aware of this strategy at the beginning of the training. (c) Percentage of reward achieved by the human participants during training.



(a) (b) (c)

Figure 4. Results of the Cooperative Ball Maze experiment. (a) Evaluation of performances of the two sub-skills of all participants. The marker styles correspond to the sub-skill preferences of the participants. (b) Evaluation performances. The error bars correspond to the 95%CIs. (c) Improvements after 20 interactions. The error bars correspond to the 95%CIs. The students improve more under student-aware teachers than both fully-assistive and random teachers (with one-sided p-values 0.069 and 0.039).

in evaluation using a trained model. Each participant was trained for 5 games and then evaluated for 1 game.

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. We define two sub-skills *leading the rotation* and *following the rotation*. We recruited N=21 (10 females and 11 males) participants to carry out human-subject experiments. The participants were first evaluated in the two sub-skills, then trained for 20 interactions, and finally evaluated in the two sub-skills again. Details can be found in the supplementary materials.

5.2 Results

A fully-assistive teacher impedes human's acquisition of skills. In the Overcooked-AI experiment shown in Figure 3(a), we observe that the students trained with a fully-assistive teacher perform worse than the students with a random teacher: it seems that a student becomes "lazy" and free rides the teacher when the teacher unilaterally adapts to the student and performs optimally. We further investigate the learning pattern of the "lazy student" problem and find out that *this "laziness" does not lie in the student's reluctance to take actions, but rather in the lack of motivation to explore and improve*. In Figure 3(c), we show the percentage of reward achieved by the student in Overcooked-AI during training. Compared with the student-aware counterpart, the percentage of reward achieved by humans is similar. However, only 17% of the participants of the group find out the efficient strategy (Figure 3(b)), which is crucial to achieving high scores in the evaluation.

Partially assistive or random partner motivates students to explore new strategies. By leaving some/all work to the student, partially assistive and random teachers both motivate the student to acquire new skills. This is shown in Figure 3(b) that most of the students under these two teachers can find out the efficient strategy in Overcooked-AI. However, their performance and the robustness of the learned strategies differ significantly. Though multiple explanations could account for it, we hypothesize the student under the random teacher learns a single fixed strategy to finish the task alone (Figure 3(c)). Such a strategy that completes the task alone cannot utilize the possibly helpful inputs from the partner, therefore resulting in a poorer performance score.

An individualized curriculum should be designed for the student. In the post-experiment survey of Cooperative Ball Maze, we asked the participants "which mode of the robot is easier to cooperate with?". Out of the 21 participants, 4 participants preferred to follow the robot and 17 participants preferred to lead the robot. Moreover, as we evaluated the student performance with partners of different sub-skills, we found that the student performances were consistent with their declared preferences (Figure 4(a)). That is to say, the student may have a bias over which strategy to acquire, and tailoring the teaching curriculum to focus on that specific strategy is efficient and more intuitive to the student. As demonstrated in Figure 5(a), after the first 6 trials that estimated the student's proficiency for each sub-skill, the teacher found out this student improved more as the leader, therefore, the teacher allocated 10 trials to perfect the *leading* sub-skills and only 4 trials for *following*. Moreover, one participant in the random teacher group responded "the robot leading mode is too difficult and I gave up". This demonstrates the importance of an individualized curriculum: though there are multiple equally optimal strategies, the individual may have strong preferences, and teaching a non-preferable strategy will discourage the student from learning anything at all. We refer the readers to the Appendix for the complete data of all participants.

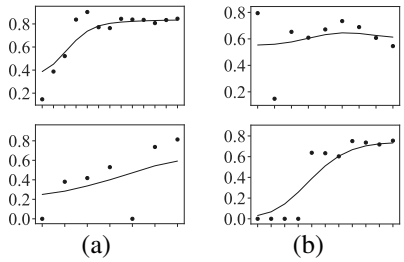


Figure 5. Sub-skill performances (vertical axis) with respect to training progress (horizontal axis) of two example participants trained by the student-aware teacher. Dots represent the raw scores and lines represent the smoothed scores. The top and bottom figures correspond to leading and following sub-skills respectively. (a) Participant 4. The student improved more when trained in the leading sub-skill. (b) Participant 6. The student improved more when trained in the following sub-skill.

6 Limitation

Decomposition into sub-skills. For many tasks, it is not easy to identify distinct roles to fulfill the local-independence criteria of sub-skills. We manually decompose the skill into a few sub-skills according to the role of the student. Often, such a decomposition may not be possible or requires careful design. We can mitigate this problem with recent progress on skill decomposition in single-agent task [44] and role-based task decomposition in multi-agent tasks [62]. However, it still demands much more effort to verify their efficacy with a real human on real-world tasks.

Teacher's Knowledge. In the definition of the teaching task, we assume the teacher has full knowledge of optimal policies. However, it can be hard for the robot to know the oracle human policy beforehand. To make the conceptual framework practical, we need to relax the requirement on the teacher's prior knowledge. In our implementation, we reduce such an assumption by approximating the distance to the optimal policy by the difference in performances. There can be cases where the optimal performance is hard to know or such relaxation results in severe information loss. We need more insights on tasks to make the framework practical.

Curriculum design. In this work, we only design the curriculum over different sub-skills. However, during our experiment, we observe that humans show various responses to the same sub-skill of different difficulties. As a result, a finer-grained curriculum on the sub-skill training shall be found to further facilitate human learning.

7 Conclusion

In this work, we propose a conceptual framework, Cooperative Robot Teaching, that enables robots to teach humans in cooperative tasks. We show that, by abstracting a teaching task over the original duo cooperative task, the robot can learn to act as a specialized teacher to humans. To be more specific, we model the teaching task as a POMDP with hidden student policy and propose a partially assistive teaching curriculum to support human learning. We believe that robot teaching fills in the gap in the bilateral knowledge transfer in HRI: unlike other HRI tasks where the humans instruct the robots how to behave, the role is reversed and robots try to instill the knowledge back into humans. Despite the challenges that lie ahead, we believe that robot teaching has great potential and is a necessary step forward to bring robots closer to our daily life.

Acknowledgements. This research is supported in part by the National Research Foundation, Singapore under its Medium Sized Centre Program, Center for Advanced Robotics Technology Innovation (CARTIN), and AI Singapore Programme (AISG Award No: AISG2-PhD-2022-01-036[T] and AISG2-PhD-2021-08-014), and by the Science and Engineering Research Council, Agency of Science, Technology and Research, Singapore, under the National Robotics Program (Grant No. 192 25 00054).

References

[1] R. M. Gillies. Cooperative learning: Review of research and practice. *Australian Journal of Teacher Education*, 41:39–54, 2016.

[2] J. A. Ross and E. M. Smyth. Differentiating cooperative learning to meet the needs of gifted learners: A case for transformational leadership. *Journal for the Education of the Gifted*, 19: 63–82, 1995.

[3] S. Omidshafiei, D. K. Kim, M. Liu, G. Tesauro, M. Riemer, C. Amato, M. Campbell, and J. P. How. Learning to teach in cooperative multiagent reinforcement learning. *AAAI Conference on Artificial Intelligence*, 2019.

[4] E. García and E. Weiss. The teacher shortage is real, large and growing, and worse than we thought. the first report in" the perfect storm in the teacher labor market" series. *Economic Policy Institute*, 2019.

[5] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016.

[6] A. Coates, P. Abbeel, and A. Y. Ng. Apprenticeship learning for helicopter control. *Communications of the ACM*, 52(7):97–105, 2009.

[7] F. Suárez-Ruiz, X. Zhou, and Q.-C. Pham. Can robots assemble an ikea chair? *Science Robotics*, 3(17), 2018.

[8] S. Nikolaidis and J. Shah. Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy. In *ACM/IEEE International Conference on Human-Robot Interaction*, 2013.

[9] M. Chen, H. Soh, D. Hsu, S. Nikolaidis, and S. Srinivasa. Trust-aware decision making for human-robot collaboration: Model learning and planning. *ACM Transactions on Human-Robot Interaction*, 9(2):1–23, 2020.

[10] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan. Cooperative inverse reinforcement learning. In *Advances in Neural Information Processing Systems*, 2016.

[11] R. Hambleton and H. Swaminathan. *Item Response Theory: Principles and Applications*. Evaluation in education and human services. 2013.

[12] L. Jones. *The Student-centered Classroom*. 2007.

[13] S. Nikolaidis and J. Shah. Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy. *ACM/IEEE International Conference on Human-Robot Interaction*, pages 33–40, 2013.

[14] A. D. Dragan and S. S. Srinivasa. A policy-blending formalism for shared control. *International Journal of Robotics Research*, 32(7):790–805, 2013.

[15] S. Reddy, A. D. Dragan, and S. Levine. Shared autonomy via deep reinforcement learning. In *Robotics: Science and Systems*, 2018.

[16] J. Leike, D. Krueger, T. Everitt, M. Martic, V. Maini, and S. Legg. Scalable agent alignment via reward modeling: a research direction. *CoRR*, 2018.

[17] H. J. Jeon, S. Milli, and A. Dragan. Reward-rational (implicit) choice: A unifying formalism for reward learning. In *Advances in Neural Information Processing Systems*, 2020.

[18] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 2017.

[19] A. Fern, S. Natarajan, K. Judah, and P. Tadepalli. A decision-theoretic model of assistance. *Journal of Artificial Intelligence Research*, 50:71–104, 2014.

[20] R. Shah, P. Freire, N. Alex, R. Freedman, D. Krasheninnikov, L. Chan, M. D. Dennis, P. Abbeel, A. Dragan, and S. Russell. Benefits of assistance over reward learning, 2021.

[21] O. Macindoe, L. Pack Kaelbling, and T. Lozano-Pérez. Pomcop: Belief space planning for sidekicks in cooperative games. *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2021.

[22] B. J. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[23] S. Nikolaidis, A. Kuznetsov, D. Hsu, and S. Srinivasa. Formalizing human-robot mutual adaptation via a bounded memory based model. In *ACM/IEEE International Conference on Human Robot Interaction*, 2016.

[24] S. Nikolaidis, D. Hsu, and S. Srinivasa. Human-robot mutual adaptation in collaborative tasks: Models and experiments. *International Journal of Robotics Research*, 36(5-7):618–634, 2017.

[25] S. Nikolaidis, Y. X. Zhu, D. Hsu, and S. Srinivasa. Human-robot mutual adaptation in shared autonomy. In *ACM/IEEE International Conference on Human-Robot Interaction*, 2017.

[26] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *International Conference on Machine Learning*, 2009.

[27] X. Zhu. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. *AAAI Conference on Artificial Intelligence*, 2015.

[28] J. Liu, X. Zhu, and H. Ohannessian. The teaching dimension of linear learners. In *International Conference on Machine Learning*, 2016.

[29] S. Mei and X. Zhu. Using machine teaching to identify optimal training-set attacks on machine learners. In *AAAI Conference on Artificial Intelligence*, 2015.

[30] F. Khan, B. Mutlu, and J. Zhu. How do humans teach: On curriculum learning and teaching dimension. In *Advances in Neural Information Processing Systems*, 2011.

[31] W. Liu, B. Dai, A. Humayun, C. Tay, C. Yu, L. B. Smith, J. M. Rehg, and L. Song. Iterative machine teaching. In *International Conference on Machine Learning*, 2017.

[32] I. Gur, N. Jaques, Y. Miao, J. Choi, M. Tiwari, H. Lee, and A. Faust. Environment Generation for Zero-Shot Compositional Reinforcement Learning. *Advances in Neural Information Processing Systems*, 2022.

[33] R. Portelas, C. Colas, K. Hofmann, and P.-Y. Oudeyer. Teacher algorithms for curriculum learning of Deep RL in continuously parameterized environments. *Annual Conference on Robot Learning*, 2019.

[34] M. Fontaine*, Y.-C. Hsu*, Y. Zhang*, B. Tjanaka, and S. Nikolaidis. On the Importance of Environments in Human-Robot Coordination. *Robotics: Science and Systems*, 2021.

[35] D. S. Brown and S. Niekum. Machine teaching for inverse reinforcement learning: Algorithms and applications. *AAAI Conference on Artificial Intelligence*, 2019.

[36] B. P. Gerkey and M. J. Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954, 2004.

[37] J. P. González-Brenes and J. Mostow. What and when do students learn? fully data-driven joint estimation of cognitive and student models. In *Educational Data Mining*, 2013.

[38] D. K. Kim, M. Liu, S. Omidshafiei, S. Lopez-Cot, M. Riemer, G. Habibi, G. Tesauro, S. Mourad, M. Campbell, and J. P. How. Learning hierarchical teaching policies for cooperative agents. *International Joint Conference on Autonomous Agents and Multiagent Systems*, 2020.

[39] N. Jaques, A. Lazaridou, E. Hughes, C. Gulcehre, P. A. Ortega, D. J. Strouse, J. Z. Leibo, and N. de Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. *International Conference on Machine Learning*, 2019.

[40] A. N. Rafferty, E. Brunskill, T. L. Griffiths, and P. Shafto. Faster teaching by pomdp planning. In *Artificial Intelligence in Education*, pages 280–287, 2011.

[41] A. Singla, I. Bogunovic, G. Bartók, A. Karbasi, and A. Krause. Near-optimally teaching the crowd to classify. *International Conference on Machine Learning*, 2014.

[42] S. Zilles, S. Lange, R. Holte, and M. Zinkevich. Models of cooperative teaching and learning. *Journal of Machine Learning Research*, 12:349–384, 2011.

[43] T. Doliwa, H. U. Simon, and S. Zilles. Recursive teaching dimension, learning complexity, and maximum classes. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010.

[44] M. Srivastava, E. Biyik, S. Mirchandani, N. Goodman, and D. Sadigh. Assistive teaching of motor control tasks to humans. In *Advances in Neural Information Processing Systems*, 2022.

[45] A. Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. Studies in computational intelligence. 2012.

[46] K. Shiarlis, M. Wulfmeier, S. Salter, S. Whiteson, and I. Posner. TACO: Learning task decomposition via temporal alignment for control. In *International Conference on Machine Learning*, 2018.

[47] T. Kipf, Y. Li, H. Dai, V. Zambaldi, A. Sanchez-Gonzalez, E. Grefenstette, P. Kohli, and P. Battaglia. Compile: Compositional imitation learning and execution. In *International Conference on Machine Learning*, 2019.

[48] P. Evrard and A. Kheddar. Homotopy switching model for dyad haptic interaction in physical colåålaborative tasks. In *Joint Euro Haptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2009.

[49] A. Kheddar. Human-robot haptic joint actions is an equal control-sharing approach possible? In *International Conference on Human System Interactions*, pages 268–273, 2011.

[50] N. Jarrassé, T. Charalambous, and E. Burdet. A framework to describe, analyze and generate interactive motor behaviors. *Plos One*, 7(11):1–13, 11 2012.

[51] A. Mörtl, M. Lawitzky, A. Kucukyilmaz, M. Sezgin, C. Basdogan, and S. Hirche. The role of roles: Physical cooperation between humans and robots. *The International Journal of Robotics Research*, 31(13):1656–1674, 2012.

[52] 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.

[53] 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.

[54] J. Gonzalez-Brenes, Y. Huang, and P. Brusilovsky. General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge. In *Educational Data Mining*, 2014.

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

[56] 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.

[57] M. Carroll, R. Shah, M. K. Ho, T. L. Griffiths, S. A. Seshia, P. Abbeel, and A. D. Dragan. On the utility of learning about humans for human-ai coordination. In *Advances in Neural Information Processing Systems*, 2019.

[58] P. Knott, M. Carroll, S. Devlin, K. Ciosek, K. Hofmann, A. D. Dragan, and R. Shah. Evaluating the robustness of collaborative agents. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, 2021.

[59] R. Charakorn, P. Manoonpong, and N. Dilokthanakul. Investigating partner diversification methods in cooperative multi-agent deep reinforcement learning. In *Iconip*, 2020.

[60] P. Nalepka, J. Gregory-Dunsmore, J. Simpson, G. Patil, and M. Richardson. Interaction flexibility in artificial agents teaming with humans. In *Annual Meeting of the Cognitive Science Society*, 2021.

[61] B. Sarkar, A. Talati, A. Shih, and S. Dorsa. Pantheonrl: A marl library for dynamic training interactions. In *AAAI Conference on Artificial Intelligence (Demo Track)*, 2022.

[62] T. Wang, T. Gupta, A. Mahajan, B. Peng, S. Whiteson, and C. Zhang. {RODE}: Learning roles to decompose multi-agent tasks. In *International Conference on Learning Representations*, 2021.