Zero-shot linear combinations of grounded social interactions with Linear Social MDPs

Anonymous Author(s)
Affiliation
Address
email

Abstract: Humans and animals engage in rich social interactions. It is often theorized that a relatively small number of basic social interactions give rise to the full range of behavior observed. But no computational theory explaining how social interactions combine together has been proposed before. We do so here. We take a model, the Social MDP, which is able to express a range of social interactions, and extend it to represent linear combinations of social interactions. Practically for robotics applications, such models are now able to not just express that an agent should help another agent, but to express goal-centric social interactions. Perhaps an agent is helping someone get dressed, but preventing them from falling, and is happy to exchange stories in the meantime. How an agent responds socially, should depend on what it thinks the other agent is doing at that point in time. To encode this notion, we take linear combinations of social interactions as defined in Social MDPs, and compute the weights on those combinations on the fly depending on the goals of other agents. This new model, the Linear Social MDP, enables zero-shot reasoning about complex social interactions, provides a mathematical basis for the long-standing intuition that social interactions should compose, and leads to interesting new behaviors that we validate using human observers. Complex social interactions are part of the future of robotics, and having principled mathematical models built on a foundation like MDPs will make it possible to bring social interactions to every robotic application.

Keywords: social interaction, goal selection, planning

1 Introduction

Machines are only able to understand and reproduce a fairly small and stilted part of the rich social behaviors that we observe humans and animals engage in. This is in part because much of the work on social robotics is based on ad-hoc approaches rather than mathematical models of social interactions, and in part because of an assumption that a relatively small number of basis social interactions will eventually give rise to the rich behavior we observe in the animal kingdom. Exactly what combining social interactions together means mathematically is left unsaid in such cases. We propose a model for social interactions and demonstrate it on a simulated robot that both has a mathematical definition for what social interactions are, and, for the first time, defines what linear combinations of social interactions are. This gives rise to complex behaviors enabling the robot to have relationships that depend on mutual goals, for example, helping an agent achieve some goals, while being willing to exchange favors to achieve another set of goals, while preventing the other agent from doing something troublesome.

We make the following contributions, 1. Linear Social MDPs, see Fig. 1, which allow robots to zero-shot carry out combinations of social interactions that respond on the fly as the goals of other agents change, 2. a demonstration of Linear Social MDPs in a grid world, see Fig. 2 for an example, and 3. validation of the resulting behaviors that show humans can recognize them as social.

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

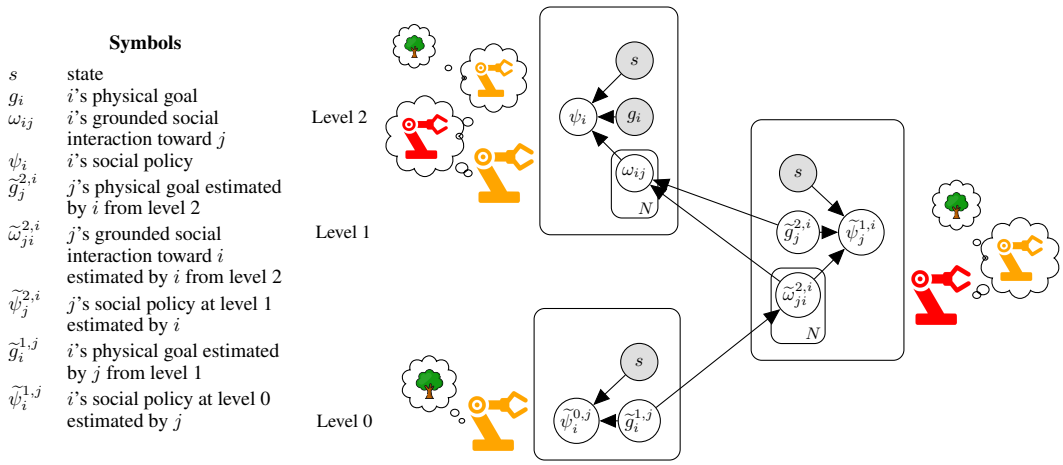


Figure 1: A yellow robot, i, performing nested inference about social interactions with a red robot, j. A level 0 agent is an MDP; a level 1 agent has social goals, but reasons about other agents as if they are level 0, so they don't have social goals. Here, the yellow agent is a level 2 agent; it considers any social interactions that the red agent might have. This is the basic setup for a Social MDP, with a critical difference – agents compute the goals of another agent, g, and then compute the compatibility between those goals and a set of N social interactions, ω, they wish to engage in. The social behavior of the robots is conditioned on the goals they believe the other agent has.

2 Related Work

Most research on social robotics is carried out without a model of what social interactions are [1]. We propose such a model that gives rise to complex social behaviors. In general, we believe that mathematical models for social interactions that are understandable from the perspective of robotics and compose with common robotic frameworks like MDPs, will both shed light on what social interactions are, and bring social robotics into the mainstream.

Several types of models have been explored to enable agents to effectively interact with one another. Inspired by cognitive science, theory-of-mind-based models [2, 3, 4, 5, 6] and Bayesian inverse planning [7, 8] approaches are used for goal inference. In reinforcement learning, methods like learning reward functions of other agents [9] and learning a latent representation of other agents' strategies [10] are used to cooperate with another agent. These methods mainly consider interactions that are cooperation or conflict.

Social MDPs [11, 12] similarly estimate another agent's reward function but this estimation is performed recursively by solving MDPs at different levels. This recursive estimate enables a robot to perform social interactions by considering the other agent's social behaviors. Our model extends Social MDPs to change their social interactions to adapt to another agent's goals.

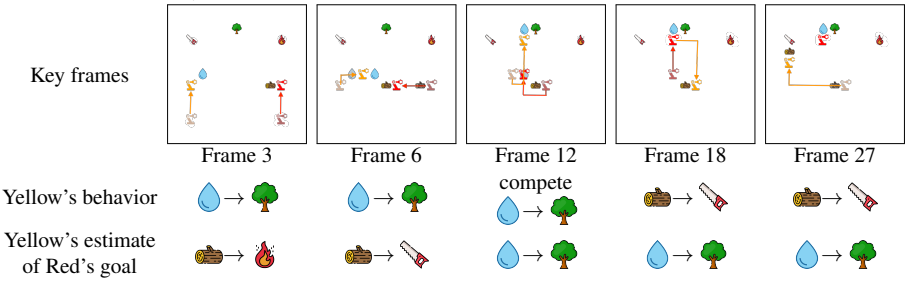
Prior research on goal or task selection includes using symbolic planners [13], a situation model [14], or task relevancy [15]. These approaches require understanding about knowledge of the tasks or planning domains. In multiagent settings, game theoretic approaches such as fictitious play [16] have been applied in coordination [17, 18] and trajectory forecasting [19] scenarios to select interaction strategies. Approaches such as selecting policy based on other agents' goals [20] or planning by finding equilibria [21] also consider what other agents may want to do in action selection. Our model also considers other agents' goals, but use it for social interactions. The combinations of social interactions formulated in our model can respond to changes in the goals of other agents in manner which no prior work has could before.

3 Linear Social MDPs

Our model extends Social MDPs [11, 12] to condition the social goal on the physical goal of another agent. Social MDPs operate by encoding social interactions in the reward function of an MDP. Agents

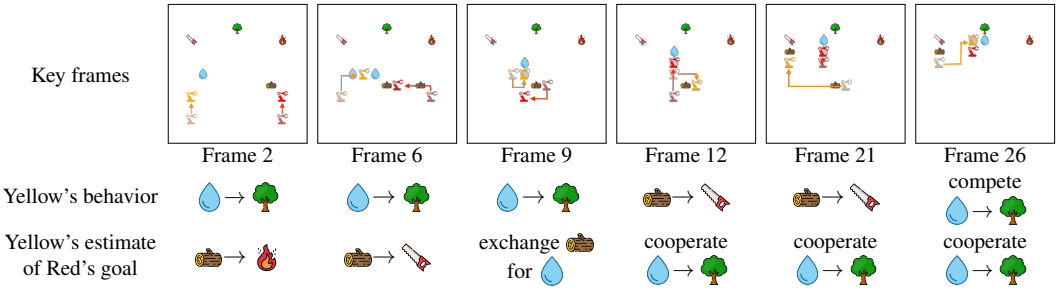
The physical and social goals for the two robots are the same for each example:
Yellow's goals → , → , compete for →
Red's goals → , → , exchange for , cooperate →

(a) Yellow: Level 1, Red: Level 0 Video: https://linear-social-mdp.github.io/scenarios/scenario-77/#level-1



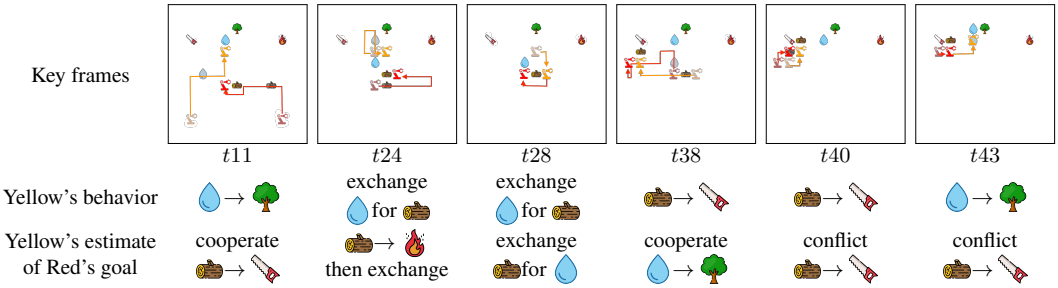
Yellow brings water to the tree while the Red is pushing the wood. As Red approaches Yellow briefly competes with Red to hurriedly move the water away.

(b) Yellow: Level 2, Red: Level 1 Video: https://linear-social-mdp.github.io/scenarios/scenario-77/#level-2



Rather than competing, Yellow and Red cooperate. This is possible because at Level 2, Yellow can recognize that Red has the capacity to cooperate, and Red has the ability to try to cooperate by exchanging objects. Neither of these are possible in (a) as neither agent is social enough.

(c) Yellow: Level 3, Red: Level 2 Video: https://linear-social-mdp.github.io/scenarios/scenario-77/#level-3



As the level of both agents goes up, they become more social. Yellow thinks that Red is constantly attempting to respond socially to it, leading to both more cooperative behavior and more conflicts toward the end.

Figure 2: Three scenarios starting from the same initial conditions, with the same two robots, having the same goals (shown at the top), and the same five objects/locations (, , , , and). Each time, both robots are reasoning at different levels of recursion; deeper levels of recursion lead to more complex behavior as they assume other agents are more social. The graphical model for the first two levels of social reasoning are shown in Fig. 1. First we show key frames from videos of the robot's behavior, then we show what physical and social goals the robots had at various times. We then provide a brief description of what the robots did. Note the increasingly complex behavior at deeper levels of social reasoning. Full videos for all the scenarios with results are available in our online appendix https://linear-social-mdp.github.io/scenarios/

l	Levels of recursive reasoning
s^t	Observed state at time t
a_i^t, a_j^t	Actions for agent i and j at time t
g_i	i's physical goal
Ω_{ij}	i's social goal toward j
ω_{ij}	i's each grounded social interaction toward j in social goal Ω_{ij}
ψ_i^l	i's social policy computed at level l
$\widetilde{g}_j^{l,i,t}$	j's physical goal estimated by i from level l at time t
$\widetilde{\Omega}_{ji}^{l,i,t}$	j's social goal toward i estimated by i from level l at time t
$\widetilde{\omega}_{ji}^{l,i,t}$	j's each grounded social interaction toward i in social goal $\widetilde{\Omega}_{ji}^{l,i,t}$ estimated by i from level l at time t
$\psi_j^{l-1,i}$	j's social policy at level $l-1$ estimated by i
R_i^l	i's reward function at level l
$r(s, g_i)$	i's reward for physical goal g_i
$R_{\Omega_{ij}}^l$	i's social reward toward j at level l
$c(a_i)$	Cost for taking action a_i
$g_{\omega_{ij}}$	Physical goal involved in the grounded social interaction ω_{ij}
$\xi_{\omega_{ij}}$	Type of social interaction involved in the grounded social interaction ω_{ij}
Q_i^l	i's state value function at level l

To compute the policy ψ_i^l:

Require: $l, s^t, a_i^t, a_j^t, \Omega_{ij}, g_i$
  if $l = 0$ **then**
    solve MDP for agent i
  else
    $\widetilde{g}_j^{l,i,t} \leftarrow$ sample $P(\widetilde{g}_j^{l,i,t} | s^{1:t-1})$
    $\widetilde{\Omega}_{ji}^{l,i,t} \leftarrow$ compute $P(\widetilde{\omega}_{ji}^{l,i,t} | s^{t-1}, a_i^{t-1}, a_j^{t-1})$
    $\widetilde{\psi}_j^{l-1,i} \leftarrow \widetilde{\psi}_j^{l-1,i}(s^t, a_j^t, a_i^t, \widetilde{\Omega}_{ji}^{l,i,t}, \widetilde{g}_j^{l,i,t})$
    compute $R_i^l(s^t, a_i^t, a_j^t, \Omega_{ij}, g_i)$
    compute $Q_i^l(s^t, a_i^t, a_j^t, \Omega_{ij}, g_i, \widetilde{\psi}_j^{l-1,i})$
    $\psi_i^l \leftarrow \text{argmax}_{a_i \in \mathcal{A}_i} Q_i^l$
  end if

Figure 3: (left) A gloss of the key notation used. (right) The algorithm to solve Linear Social MDPs at each time step. We use the estimated social policy $\widetilde{\psi}_j^{i,l-1}$ at the previous time step to update the estimated rewards. At $t = 0$ goals are sampled uniformly.

estimate what another agent is doing, i.e., their reward function, then incorporate that reward function into their own. How they incorporate other agent's reward functions determine what social interaction takes place. Incorporating the reward of another agent directly ensures that the two agent's incentives are aligned and they are likely to help one another. Doing so with the opposite sign ensures that the agent will try to minimize the reward of another agent, appearing to conflict. Reasoning is nested, where agents can be social toward agents they consider asocial (level 1 reasoning), or toward agents that they assume will also be social (level 2 reasoning). Deeper levels of reasoning allow for more complex social inferences.

Social MDPs have a major drawback: they can only encode one social interaction regardless of what the other agent is doing. An agent that is being helpful will always be helpful, even if the other agent is doing something harmful, this is an unrealistic and unreasonable limitation for real-world robotics. We create Linear Social MDPs to overcome this problem by allowing for linear combinations of social interactions where the coefficients of the interaction depend on the estimated goals of another agent. The degree to which the other agent's goals align with any one social interaction determine how strongly it will be incorporated into an agent's reward function. As a result, agents can go from being helpful, to being asocial, to being unhelpful, etc. in the course of a short interaction.

A Linear Social MDP for an agent i interacting with agent j at level, l, is defined as:

$$M_i^l = \langle \mathcal{S}, \mathcal{A}, T, \Omega_{ij}, g_i, R_i^l, \gamma \rangle \tag{1}$$

where $\mathcal{S}$ is a set of states s; $\mathcal{A} = \mathcal{A}_i \times \mathcal{A}_j$ is the set of joint actions of agents i and j; T is the probability distribution of going from state $s \in \mathcal{S}$ to next state $s' \in \mathcal{S}$ given actions of both agents: $T(s' \mid s, a_i, a_j)$; Ω_{ij} is agent i's intended social goal with agent j, it consists of a set of grounded social interactions ω_{ij}; g_i is agent i's physical goal; R_i^l is the l-th level reward function for agent i based on its estimate of other agents' rewards; and γ is a discount factor, $\gamma \in (0, 1)$.

3.1 Representing combinations of social interactions

Each $\omega \in \Omega_{ij}$ is a grounded social interaction with two components: $\xi_{\omega_{ij}}$, one of the five types of social interaction that i should carry out toward j as defined in Tejwani et al. [12]; and $g_{\omega_{ij}}$, the physical goal that i should think j is pursuing when this social interaction should be carried out. Together, these two components define a social interaction that is specific to a set of physical goals.

The overall reward of an agent i at each time step is computed as follows

$$R_i^l(s, a_i, a_j, \Omega_{ij}, g_i) = r(s, g_i) + R_{\Omega_{ij}}(g_i, s, a_i, a_j) - c(a_i) \tag{2}$$

Where Ω_{ij} is a set of social goals conditioned on physical goals, we allow for any linear combination of such, g_i is the physical goal of the current agent if any, and $c(a_i)$ is the cost of an action. Originally

rewards for Social MDPs were formulated in terms of distances between goals, but this restricted the framework to goals between which one could compute a reasonable Euclidean distance. We relax this condition here and instead compute the distance between physical goals as the shortest path from the current world state to the physical goal state, $r(s, g_i)$.

The social component of the reward function uses $\xi_{\omega_{ij}}$ to transform the estimated reward of another agent into a social behavior; see main table in Tejwani et al. [12] for a breakdown. Agent i's social reward when interacting with agent j is then

$$R^l_{\Omega_{ij}}(g_i, s, a_i, a_j) = \sum_{\omega_{ij} \in \Omega_{ij}} \int_{\widetilde{\omega}^{l,i}_{ji}} P(\omega_{ij}|s) P(\widetilde{\omega}^{l,i}_{ji} \mid s, a_i, a_j) \xi_{\omega_{ij}}(g_i, g_{\omega_{ij}}, \widetilde{\omega}^{l,i}_{ji})\, d\widetilde{\omega}^{l,i}_{ji} \tag{3}$$

This weights a social behavior $\xi_{\omega_{ij}}$ by whether that behavior is relevant to another agent's goals $g_{\omega_{ij}}$: $P(\omega_{ij}|s) \approx P(\widetilde{g}_j = g_{\omega_{ij}}|s)$, computed with Eq. (7).

3.2 Planning combinations of social interactions with Linear Social MDPs

The Q function is the sum of immediate reward and the expected value in the future by considering the estimated social policy of other agent j at a lower level l-1.

$$Q^l_i(s, a_i, a_j, \Omega_{ij}, g_i, \widetilde{\psi}^{l-1,i}_j) = R(s, a_i, a_j, \Omega_{ij}, g_i) + \gamma \sum_{s' \in S} T(s, a_i, a_j, s') V^l_i(s', \Omega_{ij}, g_i, \widetilde{\psi}^{l-1,i}_j) \tag{4}$$

We denote the estimated social policy for agent j at reasoning level $l-1$ as $\widetilde{\psi}^{l-1,i}_j : \mathcal{S} \times \mathcal{A} \times \widetilde{\Omega}^{l,i}_{ji} \times \widetilde{G}^{l,i}_j \to [0, 1]$. To compute the state-action value $V^l_i(s', \Omega_{ij}, g_i, \widetilde{\psi}^{l-1,i}_j)$, Linear Social MDPs take the expectation over the estimated goals and actions of agent j:

$$\begin{aligned} V^l_i(s', \Omega_{ij}, g_i, \widetilde{\psi}^{l-1,i}_j) &= \max_{a'_i \in \mathcal{A}_i} \left\{ E_{\widetilde{g}^{l,i}_j, \widetilde{\Omega}^{l,i}_{ji}, a'_j} [Q^l_i(s', a'_i, a'_j, \Omega_{ij}, g_i, \widetilde{\psi}^{l-1,i}_j)] \right\} \\ &= \max_{a'_i \in \mathcal{A}_i} \left\{ \sum_{a'_j \in \mathcal{A}_j} \sum_{\widetilde{g}^{l,i}_j} \int_{\widetilde{\omega}^{l,i}_{ji}} \underbrace{P(\widetilde{g}^{l,i}_j|s^{1:t})}_{\substack{\text{estimate physical goal} \\ \text{(Eq. 7)}}} \underbrace{P(\widetilde{\omega}^{l,i}_{ji} \mid s, a_i, a_j)}_{\substack{\text{estimate social goal} \\ \text{(Eq. 6)}}} \underbrace{\widetilde{\psi}^{l-1,i}_j(s', a'_i, a'_j, \widetilde{\omega}^{l,i}_{ji}, \widetilde{g}^{l,i}_j)}_{\substack{\text{estimate social policy} \\ \text{(Eq. 8)}}} Q^l_i(\cdot) d\widetilde{\omega}^{l,i}_{ji} \right\} \end{aligned} \tag{5}$$

Fig. 1 shows the overview of the model. For agent i at level l, the distributions of estimated physical goal and grounded social interaction of agent j ($\widetilde{g}^{l,i}_j$ and $\widetilde{\omega}^{l,i}_{ji}$) are further used to update the agent j's social policy so we can get the actions agent j may take. While each agent may have multiple grounded social interactions, we consider only one estimated social goal for the other agent j at each time step when solving each agent's MDP. Fig. 3 (b) summarizes the steps to compute the state-action values and select optimal actions for any level l at time step t. We first update the distribution of the estimated goals of the other agent j using the observed state and the estimated policy from the previous time step. We then sample the goals to update the policy of the other agent j and compute the reward and Q function of the target agent i.

An agent's estimate of another agent's physical and social goals at time step t and level l can be updated based on the actions performed by the agents. At $t = 0$, we use uniform distributions for physical and social goals. The social goal, estimated at time step t, is updated after actions taken by all agents at the previous time step. This update is similar to the belief update in the POMDP framework but based on the estimated social policy of the other agent j:

$$\begin{aligned} P(\widetilde{\omega}^{l,i,t}_{ji} \mid s^{1:t-1}, a^{1:t-1}_i, a^{1:t-1}_j) \propto & P(\widetilde{\omega}^{l,i,t-1}_{ji} \mid s^{1:t-2}, a^{1:t-2}_i, a^{1:t-2}_j) \\ & \sum_{\widetilde{g}^{l,i,t-1}_j} P(a^{t-1}_j \mid s^{t-1}, \widetilde{\omega}^{l,i,t-1}_{ji}, \widetilde{g}^{l,i,t-1}_j) \times T(s^{t-1}, a^{t-1}_i, a^{t-1}_j, s^t) \end{aligned} \tag{6}$$

The physical goal g_j of agent j is estimated by agent i as follows. It is marginalized over the estimated grounded social interaction as the agent is estimating the social goal at the same time.

$$P(\widetilde{g}^{l,i,t}_j|s^{1:t-1}) \propto \int_{\widetilde{\omega}^{l,i,t}_{ji}} P(s^{1:t-1}|\widetilde{g}^{l,i,t}_j, \widetilde{\omega}^{l,i,t}_{ji}) P(\widetilde{g}^{l,i,t}_j) P(\widetilde{\omega}^{l,i,t}_{ji})\, d\widetilde{\omega}^{l,i,t}_{ji} \tag{7}$$

The social policy $\widetilde{\psi}_j^{l-1,i}$ of the agent j at level l-1 is predicted by i using the Q-function at level l-1:

$$\widetilde{\psi}_j^{l-1,i}(s, a_i, a_j, \widetilde{\Omega}_{ji}^{l,i}, \widetilde{g}_j^{l,i}) = \text{Softmax}(Q_j^{l-1}(s, a_i, a_j, \widetilde{\Omega}_{ji}^{l,i}, \widetilde{g}_j^{l,i}, \widetilde{\psi}_i^{l-2,j})) \tag{8}$$

This is a softmax policy where we use a temperature parameter τ to control how much the agent j follows greedy actions. As shown in Eq. (5), in order to use agent j's Q function at level l-1, it requires to compute agent i's Q function at level l-2, and so on. Recursively solving Linear Social MDPs eventually bottoms out in level 0 where one solves an MDP.

4 Results

Given the behavior produced by the Linear Social MDP, we wanted to understand if humans could recognize the social interactions being carried out. Unlike the original Social MDPs where interactions were fixed, here the interactions changed over the duration of the scenario as the agents switch between goals. Additionally, we wanted to understand if Linear Social MDPs can recognize these social interactions, not just produce them, and to what extent other baseline models could determine what social interactions were being carried out.

Environment We use a two-agent (a yellow and red robot) 10x10 grid-world environment, with five actions (move in one of four directions or stay in place), three physical goals (watering the tree, adding logs to a fire and sawing logs), three locations (tree, fire, and saw), and two objects (a log, and a water can). In addition to the three physical goals, any combination of physical goals is possible, along with one of five social goals (cooperation, conflict, competition, coercion, or exchange) each related to one or more physical goals. Robots can move objects by pushing them.

In all experiments each robot always attempts to achieve two physical goals while engaging in social interactions relative to those goals. Those social interactions are conditioned on the physical goals of the other agent; or rather, on what the first agent thinks the second agent is doing. Despite having a fully-observable environment, agents do not have access to each other's internal states and must estimate each other's goals.

We explored every social scenario in this environment[1]. The Yellow robot always had at most one social interaction, while the Red robot always had at most two social interactions. This resulted in $6 * 6 * 5 = 180$ scenarios (eliminating the cause where neither agent considers any social interaction).

Performance A new solver for Linear Social MDPs was implemented in C++ and CUDA to perform GPU-accelerated value iteration. On a workstation with an RTX3090 updating the value estimates in parallel over 10^9 states takes about one minute. With 50 iterations, level 1 Linear Social MDPs takes about 40s, while level 3 Linear Social MDPs takes about 10 minutes.

Baselines We compared our model with inverse planning [7] and a time series classifier.

We used Bayesian inverse planning [7, 8] to infer agents' goals, given observations of their behavior. The state reward function induced by a social goal depends on the cost of another agent's action, as well as the reward function of the other agent that it wants to interact with. The other agent j's reward function was defined to be the difference of the expectation of i's reward function and j's action cost function. The scaling of the expected reward of state $\mathcal{S}$ for agent i, which determined how much j cared about i relative to its own costs. For cooperative agents, the scale was positive, and for conflicting agents, the scale is negative.

The classifier is based on concatenated features from each frame of each video [8, 13]. We built a feature vector for each robot consisting of their coordinates, distance to each resource, and whether the robot is at the goal state. These features were then input to an LSTM, the final state of which was decoded into one of the five interactions.

[1] All scenarios with detailed results for all experiments and models are available on our website https://linear-social-mdp.github.io

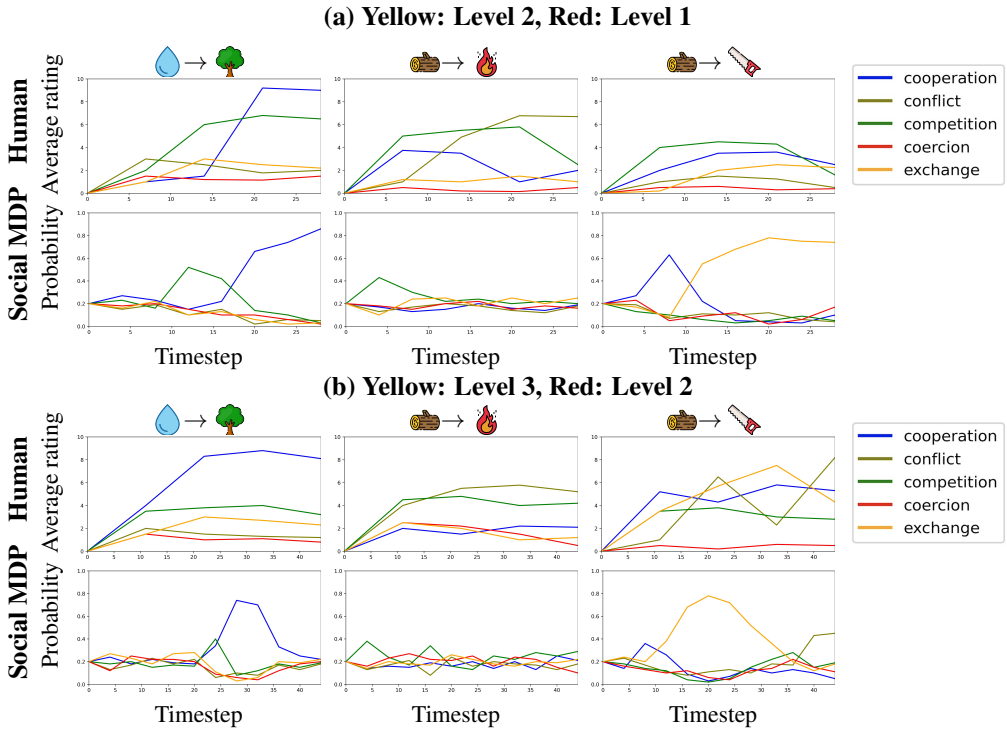


Figure 4: Humans and Linear Social MDPs were asked to predict the social interaction in each scenario at every time step. This is the result for the Yellow estimating the Red in the scenarios shown in Fig. 2. For Social MDPs, we show the probability of the grounded social interactions conditioned on each potential physical goal ($P(\tilde{\omega}_{ji}^{l,i}|s, a_i, a_j, g_{\omega_{ji}})$).

Human experiments A web interface was used to present videos of the robots engaging in social interactions and presented to subjects on Prolific [22]. Subjects were first shown several examples of each social interaction. Then, they were presented videos of social interactions and asked to classify the physical goal of a target robot (one out of three forced choice), to classify any social interactions related to that physical goal (one out of five forced choice), and to then rate their confidence. Videos were selected randomly and shown four times, incrementally revealing more of the video (starting with 25%, then 50%, 75%, and finally showing the full video). 12 subjects (mean age 36) were paid an hourly rate of $12. On an average, each subject took 11.3 minutes to complete the experiment.

Results are summarized in Table 1. Humans were able to recognize all of the social interactions when they related to any of the physical goals with high accuracy (chance is 20%, mean accuracy was almost always above 70%). This clearly shows that the Linear Social MDPs are able to perform social interactions conditioned on specific goals. A qualitative comparison between human judgements and the model is shown in Fig. 4, full results are available on our website.

The Linear Social MDPs are themselves able to recognize the goals and social ineractions in the resulting videos. While the inverse planning model and the LSTM had much lower performance.

5 Limitations

As with many methods which directly execute MDPs inference times are slow and don't scale well. We are exploring GNN-based approximations to Social MDPs to make them practical for online inference.

Social MDPs assume a fully observable state (although, note that this doesn't include the goals/rewards, both social and physical, of other agents; these are not available and must be inferred). Social POMDPs would alleviate this problem, and while they are quite straightforward to formulate, efficient inference remains a challenge.

Social Interaction	*Human*	Linear Social MDP (Ours)	Inverse Planning	LSTM
Cooperation	0.798	0.761	0.742	0.521
Conflict	0.788	0.712	0.717	0.459
Competition	0.683	0.659	0.431	0.278
Coercion	0.808	0.784	0.323	0.172
Exchange	0.669	0.681	0.446	0.081

Table 1: Humans rated how well they could understand the social interactions produced by Linear Social MDPs. Chance is 20%, overall, they were able to recognize every social interaction, with 'exchange' being the most difficult. Linear Social MDPs could recognize the resulting movies as well, while the inverse planning-based model and the LSTM had difficulty doing so. SocialMDPs produce videos that are understandable to humans, and they can recognize such videos even when other models can't.

A fundamental unknown is the contents and size of the basis space of social interactions and the set of operators that combine social interactions. There are no known methods to determine what space of the full range of social interactions that humans and animals engage in these methods can account for. Even categorizing or recognizing social interactions remains challenging. We are working on using these methods to parse videos of social interactions, not just generate behaviors, as a step in this direction.

When Social MDPs engage in an interaction, there is no guarantee that they will display the full range of what humans would recognize as that social interaction. Indeed, Social MDPs are formulated cannot help by providing information to an agent, since they don't model partial observability. At present, it is unclear how to measure this. It is also unclear how to validate some of the basic assumptions of Social MDPs and of Linear Social MDPs, such as the fact that there are multiple levels of social reasoning. At least in principle these features of Social MDPs are falsifiable, but we are still designing experiments that would enable us to falsify them.

We only consider interactions between pairs of agents. Moreover, we only consider systems that interact briefly and then reset, as does most work in robotics with MDPs. Finally, we consider the same specific type of social interaction as that of Social MDPs: social interactions that arise as a consequence of some social principle and can be modeled zero-shot, rather than social conventions. All societies have conventions that must be learned, like taboos, pleasantries, etc. Practically, for example, this may mean that an agent can touch some agents but not others, adding nuance to how an agent may be helped or hindered. In principle, such knowledge could be added a prior over the Social MDP being used, and indeed, one might define and discover social conventions automatically as the residual knowledge after reasoning about the principled social interaction. Being able to reason about combinations of social interactions, as we do here, is a step toward tackling such problems.

6 Conclusion

Linear combinations of social interactions are meaningful and lead to powerful new behavior. They allow MDPs to encode complex social interactions, where agents are not just broadly helping one another, but display a wide range of interactions that change in response to other agents' goals. This is encoded by making the coefficients of the linear combination depend on the goals of other agents. The resulting models engage in zero-shot social interactions as long as the underlying problem domain can be encoded as an MDP.

We are working on demonstrating Social MDPs on robots while they play physical multiplayer games with humans. Many games can be specified as MDPs, and we would like to have a plug-and-play solution where a generic ROS package can drive social behavior. We are working on lifting many of the limitations described above as well as on further human experiments to validate the approach and discover enhancements to the framework. In the long term, we hope to put social robotics on a firmer mathematical foundation as well as provide datasets and benchmarks that will make social interactions a first class citizen in machine learning and robotics.

References

[1] T. B. Sheridan. A review of recent research in social robotics. *Current opinion in psychology*, 36:7–12, 2020.

[2] C. L. Baker, N. D. Goodman, and J. B. Tenenbaum. Theory-based social goal inference. In *Proceedings of the thirtieth annual conference of the cognitive science society*, pages 1447–1452. Cognitive Science Society, 2008.

[3] C. L. Baker and J. B. Tenenbaum. Modeling human plan recognition using bayesian theory of mind. *Plan, activity, and intent recognition: Theory and practice*, 7:177–204, 2014.

[4] J. Kiley Hamlin, T. Ullman, J. Tenenbaum, N. Goodman, and C. Baker. The mentalistic basis of core social cognition: Experiments in preverbal infants and a computational model. *Developmental science*, 16(2):209–226, 2013.

[5] M. Kleiman-Weiner, M. K. Ho, J. L. Austerweil, M. L. Littman, and J. B. Tenenbaum. Coordinate to cooperate or compete: abstract goals and joint intentions in social interaction. In *CogSci*, 2016.

[6] N. Rabinowitz, F. Perbet, F. Song, C. Zhang, S. A. Eslami, and M. Botvinick. Machine theory of mind. In *International conference on machine learning*, pages 4218–4227. PMLR, 2018.

[7] C. L. Baker, R. Saxe, and J. B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

[8] T. D. Ullman, C. L. Baker, O. Macindoe, O. Evans, N. D. Goodman, and J. B. Tenenbaum. Help or hinder: Bayesian models of social goal inference. Technical report, Massachusetts Institute of Technology Department of Brain and Cognitive Sciences, 2009.

[9] D. Hadfield-Menell, A. Dragan, P. Abbeel, and S. Russell. Cooperative inverse reinforcement learning. *arXiv preprint arXiv:1606.03137*, 2016.

[10] A. Xie, D. P. Losey, R. Tolsma, C. Finn, and D. Sadigh. Learning latent representations to influence multi-agent interaction. In *Conference on Robot Learning (CoRL)*, 2020.

[11] R. Tejwani, Y.-L. Kuo, T. Shu, B. Katz, and A. Barbu. Social interactions as recursive mdps. In *Conference on Robot Learning (CoRL)*, 2021.

[12] R. Tejwani, Y.-L. Kuo, T. Shu, B. Stankovits, D. Gutfreund, J. B. Tenenbaum, B. Katz, and A. Barbu. Incorporating rich social interactions into MDPs. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

[13] T. Shu, M. Kryven, T. D. Ullman, and J. B. Tenenbaum. Adventures in flatland: Perceiving social interactions under physical dynamics. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2020.

[14] D. Vernon, J. Albert, M. Beetz, S.-C. Chiou, H. Ritter, and W. X. Schneider. Action selection and execution in everyday activities: A cognitive robotics and situation model perspective. *Topics in cognitive science*, 14(2):344–362, 2022.

[15] V. G. Santucci, G. Baldassarre, and E. Cartoni. Autonomous reinforcement learning of multiple interrelated tasks. In *2019 Joint IEEE 9th international conference on development and learning and epigenetic robotics (ICDL-EpiRob)*, pages 221–227. IEEE, 2019.

[16] G. W. Brown. Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, 13(1):374–376, 1951.

[17] C. Eksin and A. Ribeiro. Distributed fictitious play in potential games of incomplete information. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 5190–5196. IEEE, 2015.

[18] C. Eksin and A. Ribeiro. Distributed fictitious play for multiagent systems in uncertain environments. *IEEE Transactions on Automatic Control*, 63(4):1177–1184, 2017.

[19] W.-C. Ma, D.-A. Huang, N. Lee, and K. M. Kitani. Forecasting interactive dynamics of pedestrians with fictitious play. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[20] A. Mohseni-Kabir, D. Isele, and K. Fujimura. Interaction-aware multi-agent reinforcement learning for mobile agents with individual goals. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3370–3376. IEEE, 2019.

[21] M. Bowling, R. Jensen, and M. Veloso. A formalization of equilibria for multiagent planning. In *IJCAI*, pages 1460–1462, 2003.

[22] Prolific. https://www.prolific.co. Accessed: 2022-06-01.