Social Interactions as Recursive MDPs

Anonymous Author(s)
Affiliation
Address
email

Abstract: While machines and robots must interact with humans, providing them with social skills has been a largely overlooked topic. This is mostly a consequence of the fact that tasks such as navigation, command following, and even game playing are well-defined, while social reasoning still mostly remains a pre-theoretic problem. We demonstrate how social interactions can be effectively incorporated into MDPs by reasoning recursively about the goals of other agents. In essence, our method extends the reward function to include a combination of physical goals (something agents want to accomplish in the configuration space, a traditional MDP) and social goals (something agents want to accomplish relative to the goals of other agents). Our Social MDPs allow specifying reward functions in terms of the estimated reward functions of other agents, modeling interactions such as helping or hindering another agent (by maximizing or minimizing the other agent's reward) while balancing this with the actual physical goals of each agent. Our formulation allows for an arbitrary function of another agent's estimated reward structure and physical goals, enabling more complex behaviors such as politely hindering another agent or aggressively helping them. Extending Social MDPs in the same manner as I-POMDPs extension would enable interactions such as convincing another agent that something is true. To what extent the Social MDPs presented here and their potential Social POMDPs variant account for all possible social interactions is unknown, but having a precise mathematical model to guide questions about social interactions has both practical value (we demonstrate how to make zero-shot social inferences and one could imagine chatbots and robots guided by Social MDPs) and theoretical value by bringing the tools of MDP that have so successfully organized research around navigation to hopefully shed light on what social interactions really are given their extreme importance to human well-being and human civilization.

1 Introduction

Progress on modeling social interactions and giving machines social goals, such as being particularly nice to a user, is significantly hampered by the lack of theoretical models which characterize what social interactions are. Great practical progress was made in robot navigation and sensing, with the introduction of MDPs [1] and POMDPs [2]. Defining the problem clearly allowed us as a field to understand what we can model and how to do so. Until we take this same step for social interactions, they will remain on shaky ground despite their importance to virtually every interaction humans engage in.

We introduce an extension of MDPs, which we term Social MDPs. In the process, we make several assumptions. First, that agents have physical goals and social goals, and their overall reward structure is some arbitrary combination of the two, potentially accompanied by other terms. Physical goals are precisely what MDPs can already express, a of function of points in a configuration space. Social goals are a function of the estimate of the reward structure of another agent. For example, a reward that hinders another agent is some negative function of the estimated reward of that agent. Complicating matters is the fact that social rewards like beliefs can be recursive: an agent may want to help another agent help them. To model this, Social MDPs are recursive up to some depth, much like interactive POMDPs [3], I-POMDPs. Unlike I-POMDPs, Social MDPs are not recursive in terms of agent's beliefs about the state of the world. Instead, Social MDPs are recursive in terms of the

Submitted to the 5th Conference on Robot Learning (CoRL 2021). Do not distribute.

rewards of the agents. This makes Social MDPs and I-POMDPs orthogonal and complementary. Social MDPs are specifically formulated to not interfere with the standard extension from MDPs to POMDPs, making partial observability possible to include. While we do not develop a joint Social I-POMDP here, this is a reasonable extension which would cover far more of the space of social interactions, although one that is computationally challenging.

Our contributions are:

1. formulating Social MDPs where an agent's reward function is an arbitrary function of the recursive estimate of another agent's reward and a physical goal,
2. an implementation where that function is a linear transformation, which captures many notions of helping and hindering,
3. demonstrating that the model performs zero-shot social reasoning in agreement with a human subjects experiment, and
4. examples of the practical utility of recursive social reasoning,

In an anonymized online appendix[1] we fully enumerate all possible scenarios predicted by our model given an environment simple enough to allow doing so, demonstrating that it captures a diverse set of social behaviors. We also provide videos of the behavior of our model in all these scenarios.

2 Related Work

Modeling other agents In order to interact with other agents effectively, an agent must be able to reason about the goals, preferences, and beliefs of other agents [4]. Theory-based models for social goal attribution [5, 6, 7, 8, 9], Bayesian inverse planning to infer an agent's goal given the observations of their behaviors [10, 11], and learning the reward functions of other agents [12] have been explored. Prior research also tried to recognize social interactions such as waving and hugging in videos where people are involved in group activities [13, 14, 15]. These methods generally involve two separate stages [16]: 1) a social perception stage and 2) coordination or collaboration stage where agents interact. In contrast, Social MDPs constantly reevaluate the goals of other agents enabling them to adapt to changes in the plans of other agents. Moreover, MDPs tend to be more efficient than alternative formulations, even when solving them recursively. Social MDPs also allow for enumerating social situations by formally defining the space of what social interactions are, opening the doors to a more theoretical approach to social interactions.

Simulating social interactions Heider and Simmel [17] simulated agent trajectories to study social perceptions through a set of animations involving the movements of geometrical figures. Simulating agent behaviors in physics engines had been explored to collect datasets in fully observable [18, 19] and partially observable environments [20, 16]. The collected datasets are used to study social perception or build machine learning models that can recognize agents' goals. These frameworks assume that each agent has either a physical goal or a social goal. Here we consider a more general scenario where agents have both a physical goal and a social goal and the overall behavior is a blend of the two. This is a more practical setting as when one decides to help another agent one does not abandon every other concern and physical care. Our formulation allows for an arbitrary function of another agent's reward allowing for complex social interactions although we explore only a subset of the full power of Social MDPs here by considering only linear functions.

Learning to interact with other agents Interactive POMDPs [21, 22, 23] (I-POMDPs) are extensions of POMDPs that recursively model the beliefs of other agents. Social MDP and I-POMDPs are orthogonal. Social MDPs allow agents to reason recursively about other agents' reward functions while I-POMDPs allow agents to reason recursively about other agent's beliefs about the state of the world. The two could in principle be combined, but while Social MDPs require solving a modest number of additional nested MDPs, I-POMDPs require significantly nested inference, and when the two are combined the problem quickly becomes intractable. [24] propose a different type of approach that does not require nested inference: learning a low-dimensional representation of an other agent's strategy. This approach allows an agent to avoid another agent or to manipulate another agent into some mutually-beneficial behavior. Social MDPs, on the other hand, allow building the strategy of another agent directly into the reward function of an agent, enabling behaviors such

[1] See https://social-mdp.github.io

as helping or hindering regardless of what the other agent is trying to achieve. Moreover, Social MDPs are zero-shot, while this prior approach is not. From the point of view of generalization and sample-efficient robotics, a zero-shot approach is preferable, moreover, while opening up new doors for a more theoretical understanding of social interactions. We could combine Social MDPs with this prior work to build in latent representations of strategies into reward functions [25, 26] creating more efficient approximations of Social MDPs.

3 Social MDPs

Social MDPs are a recursive MDP (Markov decision process) with nested estimates of other agent's goals inspired by hierarchical models of games [27] and nested MDP that reason about the beliefs of other agents [28, 29, 30]. Like nested models such as I-POMDPs, Social MDPs have the notion of a level. A level 0 Social MDP is simply an MDP, agents reason about the map state. A level 1 Social MDP enables each agent to reason about the physical goals of other agents (those other agents are treated like level 0 agents). A level 2 Social MDP enables each agent to reason about the level 1 social goals of other agents. To perform this nested inference agents must have access to another agent's physical and social goals. These goals are computed by solving MDPs recursively at every level.

A level 0 agent can take physical actions, but cannot reason socially. A level 1 agent can take actions relative to another agent's physical goals, such as helping, hindering, stealing, etc. A level 2 agent can take actions relative to another agent's social and physical goals, such as avoiding an attempt to be hindered, recognizing that help is needed, joining in to help together. Levels deeper than 2 continue to describe meaningful interactions although we do not consider them here. It is unclear what level of recursion is required before agents exceed the social reasoning capacities of humans.

3.1 Assumptions

As we are in an MDP setting, all physical states are fully observable to both agents. Note however that non-physical state is not available to other agents. In other words, agents can observe one another at any distance and regardless of any obstacles but they cannot know what another agent wants (either physically or socially) by reading that agent's mind. Agents must infer the physical and social goals, if any, of other agents from their actions. Agents reason completely independently of one another.

In this paper, we consider only pairwise Social MDPs; indeed, we only experiment in settings where only two agents are present, rendering all other models equivalent to the pairwise one. Multiple agents in a pairwise MDP scale inference linearly at every level. N-way Social MDPs that consider social interactions between multiple agents at the same time are a trivial extension mathematically. Computationally however, n-way Social MDPs, as with any such model, scale exponentially in n.

As with many multiagent MDP settings, the experiments performed here assume that agents are optimal planners. This is not an inherent limitation of Social MDPs, but relaxing this assumption makes the nested IRL step far more difficult.

3.2 Formal definition of Social MDPs

A Social MDP for agent i with respect to all agents J consists of an arity (here we formulate the pairwise case) and a maximum level, l, and is defined as:

$$M_i^l = \langle \mathcal{S}, \mathcal{A}, T, \chi_{iJ}, R_i^l, \gamma \rangle \tag{1}$$

where

- S is a set of states in the environment where $s \in \mathcal{S}$.
- $\mathcal{A} = \mathcal{A}_i \times \mathcal{A}_J$ is the set of joint moves of all agents. $a_i \in \mathcal{A}_i$ is an action for agent i.
- T is the probability distribution of going from state $s \in \mathcal{S}$ to next state $s' \in \mathcal{S}$ given actions of all agents: $T\left(s' \mid s, a_i, a_J\right)$.
- χ_{iJ} is agent i's social goal toward every other agent in J.
- R_i^l is the lth level reward function for agent i that takes as input the state, joint actions, and its estimate of other agents' rewards.

- γ is a discount factor: $\gamma \in (0, 1)$.

Reward Each agent has its own physical goal, e.g., going to a landmark, as well a social goal, e.g., helping or hindering other agents. What enables Social MDPs to go beyond regular MDPs is recursive nature of the reward function which can be written in terms of the estimated rewards of other agents. The immediate reward of an agent i at each time step is computed as follows:

$$R_i^l(s, a_J, \chi_{iJ}, g_i) = r_i\left(s, a_i, g_i\right) + \sum_{j \in J, j \neq i} \chi_{ij}(\widetilde{R}_j^{j-1,i}(s, a_J, \widetilde{\chi}_{jJ}^{i,l}, \widetilde{g}_j^{i,l})) - c(a_i) \tag{2}$$

where J is the set of all agents, l is the level, $r(\cdot)$ is the static reward given the agent's own physical goal, g_i, $c(\cdot)$ is the cost for taking an action, $\widetilde{R}_j^{l-1,i}(\cdot)$ is the estimated reward for agent j from agent i's point of view based on the estimated social and physical goals for agent j. The reward is defined to be a constant, zero, for negative levels. χ_{ij} is the social goal, it transforms the reward of another agent into a goal that is part of the reward of the target agent. In this case, we instantiate the model with a linear transformation, so χ_{ij} is simply a reweighting of the social goal, i.e., of the estimated reward of the other agent. A negative value will have the target agent attempt to minimize the reward of another agent (this amounts to hindering that agent). A positive value corresponds to helping. Social goals can be eliminated entirely by setting this weight to zero.

Estimating physical and social goals To solve an agent's MDPs over different levels, it needs to estimate the other agent's physical and social goals. Similar to [18], the physical goal g_j of agent j is predicted by i as in

$$P(g_j|s^{1:T}) \propto \int_{\widetilde{\chi}_{ji}^i} P(s^{1:T}|g_j, \widetilde{\chi}_{ji}^{i,l}) \cdot P(g_j) \cdot P(\widetilde{\chi}_{ji}^{i,l})\, d\widetilde{\chi}_{ji}^{i,l} \tag{3}$$

Since the agent is estimating the social goal at the same time, the estimation of physical goal needs to marginalize over the estimated social goal as well. The social goal of agent i towards agent j estimated by agent k at level l is denoted as $\tilde{\chi}_{ij}^{k,l}$. In the two-player setting, k can be either agent i or j depending on which agent is making the estimation. We will describe how to update the estimate of the social goal in Section 3.4. When solving agent i's MDP at level l, this estimated social goal is further used to estimate the other agent j's social policy $\widetilde{\psi}_j^{i,l} : \mathcal{S} \times \mathcal{A}_J \times \widetilde{\chi}_{jJ}^{i,l} \rightarrow [0, 1]$.

3.3 Planning for Social MDPs

Analogous to MDPs, the state-action value is the sum of immediate reward and the expected value in the future. Since agent i is interacting with other agents $j \in J$, it needs to estimate what actions other agents are likely to take in order to compute its state-action value. Social MDPs take the expectation over the estimated social intention of agent j in the Q function to be

$$\begin{aligned} Q_i^l(s, a_i, a_J, \chi_{iJ}) &= R(s, a_i, \chi_{iJ}) + \gamma \sum_{s' \in S} T\left(s, a_i, a_J, s'\right) \\ &\sum_{a_i'} \sum_{j \in J, j \neq i} \sum_{a_j'} \int_{\widetilde{\chi}_{ji}^{i,l}} Pr(\tilde{\chi}_{ji}^{i,l} \mid s, a_J) \widetilde{\psi}_j^{i,l}(s', a_j', a_J', \widetilde{\chi}_{jJ}^{i,l-1}) Q_i^l(s', a_i', a_J', \chi_{iJ})\, d\widetilde{\chi}_{ji}^{i,l} \end{aligned} \tag{4}$$

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

$$\widetilde{\psi}_j^{i,l}(s, a_j, a_J, \widetilde{\chi}_{jJ}^{i,l-1}) = \frac{\exp(Q_j^{l-1}(s, a_j, a_J, \widetilde{\chi}_{jJ}^{i,l-1})/\tau)}{\sum_{a_j} \sum_{a_J} \exp(Q_j^{l-1}(s, a_j, a_J, \widetilde{\chi}_{jJ}^{i,l-1})/\tau)} \tag{5}$$

This is a softmax policy where τ is the temperature parameter controlling how much the agent j follows the greedy actions. Based on Eq. 4, 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. This involves solving recursive MDPs at levels $0, 1, \cdots, l$-1.

3.4 Social goal update

An agent's estimate of another agent's social intention at time step t and level l is updated based on the actions performed by the agents

$$Pr(\widetilde{\chi}_{ji}^{i,l,t} \mid s^{t-1}, a_J^{t-1}) = \beta Pr(\widetilde{\chi}_{ji}^{i,l,t-1} \mid s^{t-2}, a_J^{t-2}) \sum_{a_j^{t-1}} \sum_{\widetilde{g}_j^{i,l,t-1}} Pr\left(a_j^{t-1} \mid s^{t-1}, \widetilde{\chi}_{jJ}^{i,l,t-1}, \widetilde{g}_j^{i,l,t-1}\right) \times T\left(s^{t-1}, a_j^{t-1}, a_J^{t-1}, s^t\right) Pr(\widetilde{\chi}_{ji}^{i,l,t} \mid \widetilde{\chi}_{ji}^{i,l,t-1}, a_j^{t-1}) \tag{6}$$

where β is the normalizing constant and $Pr(\widetilde{\chi}_{ji}^{i,l,t} \mid \widetilde{\chi}_{ji}^{i,l,t-1}, a_j^{t-1})$ is the Kronecker delta function $\delta_K\left(\widetilde{a}_j^{t-1}, a_j^{t-1}\right)$. $\widetilde{a}_j^{t-1}$ is i's prediction of j's action given the estimated social intention $\widetilde{\chi}_{jJ}^{i,l,t-1}$ and a_j^{t-1} is the actual action taken by j at the time step $(t-1)$. The Kronecker delta function evaluates to 1 only when the predicted action is the same as the actual action, thereby resolving Eq. eq. (6) to:

$$Pr(\widetilde{\chi}_{jJ}^{i,l,t} \mid s^{t-1}, a_J^{t-1}) = \beta Pr(\widetilde{\chi}_{jJ}^{i,t-1} \mid s^{t-2}, a_J^{t-2}) \sum_{\widetilde{g}_j^{i,l,t-1}} Pr\left(a_j^{t-1} \mid s^{t-1}, \widetilde{\chi}_{jJ}^{i,l,t-1}, \widetilde{g}_j^{i,l,t-1}\right) \times T\left(s^{t-1}, a_j^{t-1}, a_J^{t-1}, s^t\right) \tag{7}$$

The social intention, estimated at time step t, is updated after actions taken by both the agents at each 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.

3.5 Value Iteration

We use value iteration to solve Social MDP M_i^l for agent i at level l. The value function at $(k+1)$-th update of value iteration satisfies the following Bellman backup operation:

$$Q_i^{l,k+1}(s, a_i, a_J, \chi_{iJ}) = R_i^l(s, a_i, \chi_{iJ}) + \gamma \sum_{s' \in \mathcal{S}} T\left(s, a_i, a_J, s'\right) V_i^{l,k}\left(s', \chi_{iJ}\right) \tag{8}$$

$$V_i^{l,k+1}(s, \chi_{iJ}) = \max_{a_i \in \mathcal{A}_i} \left\{ \sum_{j \in J} \sum_{a_j \in \mathcal{A}_j} \int_{\widetilde{\chi}_{ji}^{i,l}} Pr(\widetilde{\chi}_{ji}^{i,l} \mid s, a_J) \widetilde{\psi}_j^{i,l}(s', a_j', a_J', \widetilde{\chi}_{ji}^{i,l-1}) Q_i^{l,k+1}(s', a_i', a_J', \chi_{iJ})\, d\widetilde{\chi}_{ji}^{i,l} \right\} \tag{9}$$

After applying Eq. 9 iteratively, agent i's optimal action for level l, $OPT(M_i^l)$, can be obtained as:

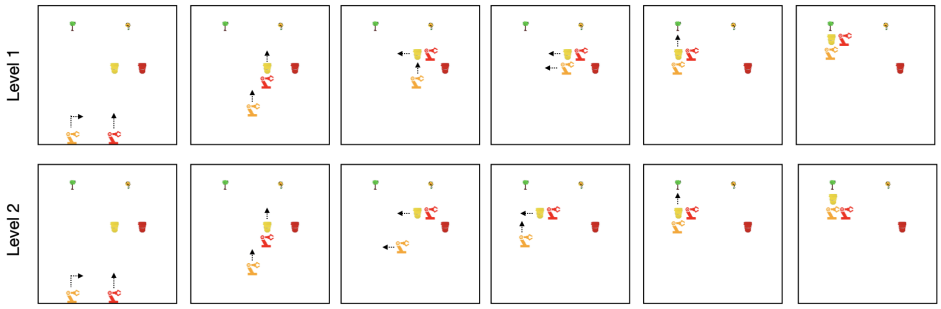
$$\operatorname*{argmax}_{a_i \in \mathcal{A}_i} \left\{ \sum_{j \in J} \sum_{a_j \in \mathcal{A}_j} \int_{\widetilde{\chi}_{ji}^{i,l}} Pr(\widetilde{\chi}_{ji}^{i,l} \mid s, a_i) \widetilde{\psi}_j^{i,l}(s', a_j', a_J', \widetilde{\chi}_{ji}^{i,l-1}) Q_i^{l,k+1}(s', a_i', a_J', \chi_{iJ})\, d\widetilde{\chi}_{ji}^{i,l} \right\} \tag{10}$$

3.6 Time Complexity

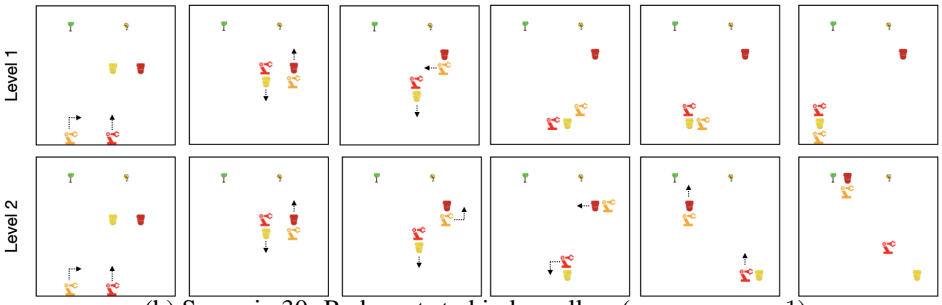
The time complexity of solving the Social MDP at level 0 is the same as that of solving an MDP. At level 1, an MDP must be solved for every agent independently in order to compute the likely physical goals of every other agent. Solving a Social MDP at level l requires solving $O((A-1)^2 l)$ MDPs, where A is the number of agents. Social MDPs form a tree with branching factor $A-1$ as every agent must compute pairwise social goal of every other agent until level 0 where the tree bottoms out. There are many potential speedups that can alleviate this runtime to allow for efficient inference even in the face of many agents. For example, a distance horizon could be used where far away agents could simply be considered non-interacting.

4 Results

We apply the Social MDP framework to a multi-agent grid world inspired by previous studies on social perceptions [11, 5, 31]. The 10×10 world consists of two agents, a yellow robot and red



(a) Scenario 28: Red wants to help yellow ($\chi_{\text{red,yellow}} = 2$)



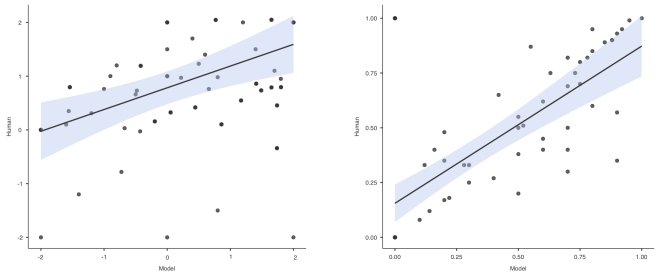
(b) Scenario 30: Red wants to hinder yellow ($\chi_{\text{red,yellow}} = -1$).

Figure 1: Two examples of zero-shot social interactions.The Social MDP gives the robots the ability to understand and predict relationships, thereby making far more efficient actions. The yellow robot wants to water the tree. Moving the yellow bucket is easy for the yellow robot, while moving the red bucket is hard for the yellow robot. The yellow robot performs inference to understand what the red robot is doing. With a level 1 Social MDP, the yellow robot assumes that the red robot has a physical goal, but not a social goal. With a level 2 Social MDP, the yellow robot assumes that the red robot has both a physical and social goal, then recursively estimates the social goal of the red robot (which is in turn modeled as a level 1 Social MDP).
(a) At level 1, the yellow robot follows the red one around. It does not understand that the red robot is trying to help. The red robot correctly executes its social goal of helping the yellow one by moving its bucket toward the tree. At level 2, the yellow robot recognizes that red is helping, projects where its future trajectory will take it, and efficiently goes to the intercept point accepting red's help.
(b) At level 1, yellow does not infer that red wants to help it. As such it attempts to move the yellow bucket and repeatedly fails, entering a local minima where the yellow bucket is the most efficient one to move without realizing that the red robot will forever prevent this. At level 2, the yellow robot recognizes that the red one is attempting to hinder it, gives up on the yellow bucket, and makes a globally-optimal move of using the harder-to-move red bucket instead.

robot, two physical landmarks, a flower and tree, and two objects, a yellow water can and red water can. The yellow agent has a low cost for moving the yellow watering can, while it has a high cost for moving the red watering can. Robots can have a physical goal of moving the water can to a target plant. Robots can have a social goal of helping or hindering to different degrees. The grid world is four connected; agents can move in any direction or choose not to move.

98 different scenarios [2] are systematically created in this grid world. Each scenario has agents as having either the same goal or different physical goals and one of 7 different scaling factors on each of their social goals (-2, -1, -0.5, 0, 0.5, 1, 2) ($2 * 7 * 7 = 98$ scenarios). 2 indicates that the social goal is weighted much more than the physical goal, and an agent wants to maximize the other agent's goal. Similarly for -2, except that an agent wants to minimize another agent's goal. At 1, agents weigh their own social goals equally with their own physical goals. At 0.5, they put twice as much weight on their physical goals as their social goals. These agents are less likely to help, particularly if helping will cost more time or energy than carrying out the physical goal. Finally, either agent can

[2] Scenarios can be viewed at https://social-mdp.github.io



(left) Weight of social goals (right) Weight physical goals

Figure 2: Twelve human subjects, and our model, the Social MDP, watched and scored 10 videos at different snapshots. These videos consist of five scenarios where robots reason at either level 1 or level 2 (presented to the users in randomized order). (left) Models and humans were asked to predict how social the agents were and the valence of the interaction (was it positive or negative). Non-social settings have a weight of 0, while adversarial settings have a social weight of -2, overwhelming the physical goal of any agent. Humans and machines predict similar social goals both in terms of value and magnitude. (right) Models and humans were asked to predict a weight factor on the physical goal, how much does this agent care about its physical goal. At 0, the physical goal is ignored. At 1, it is weighted equally with a social goal also set at 1. Human and model scores are again highly correlated. Our model is able to effectively generate trajectories that humans recognize as being social interactions. It is also able to predict the type of social interaction that humans believe occurred.

	Social MDP (ours)	Inverse Planning	Cue-based
Social Goal	**0.89**± 0.05	0.81 ± 0.02	0.23 ± 0.007
Physical Goal	**0.78**± 0.03	0.72 ± 0.04	0.08 ± 0.006

Figure 3: The coefficient of correlation between human and machine judgements for the experiment described in fig. 2. We provide two baselines and our own approach. The cue-based model is described in Shu et al. [18]. The inverse planning model is described in Ullman et al. [11]. Social MDPs produce better alignment with human judgements than other models and do not require training like the cue-based model.

have the factor set to zero; these agents only have a physical goal and no social goal. The fact that we can enumerate all possible social interactions, as predicted by our model, in a given scenario is an important missing feature of alternative representations. All 98 scenarios correspond to reasonable interactions between agents. The degree to which this is true in more complex environments and the degree to which systematically unfolding the model in more complex environments always results in what humans would describe as social interactions is an important topic for future work.

Each agent's reward for reaching its physical goal is based on that agent's geodesic distance from the goal after taking an action [11]. This physical reward function is parameterized by ρ and δ that determines the scale and shape of the physical reward: $r_i(s, a, g_i) = \max\left(\rho\left(1 - \text{distance}(s, a, g_i)/\delta\right), 0\right)$. We set the cost, c, of an action a, to 1 for grid moves and 0.1 to staying in place while ρ and δ were set to 1.25 and 5, respectively. The discount factor, γ, was set to 0.99.

Two of the scenarios selected for our experiment are shown in fig. 1. We used the Social MDP to select actions for the yellow agent which has a physical goal, watering the flower or tree, while interacting with the red agent. The red agent had a physical goal, watering the tree, and a varying social goal. At every time step, the yellow agent estimates the physical and social goal of the red agent, depending on the Social MDP level. At deeper levels the agents behave more optimally and more socially.

To quantitatively establish the quality of the social inferences made by the Social MDPs, we compare human judgements of 12 subjects against those of two baseline models: inverse planning [11] and a recent cue-based model [18]. Humans and models had to estimate the physical and social goals of agents in these environments when the agents were acting both as level one agents (unaware that the other agents are also social) and as level two agents (who could account for the fact that the other agents are social). In fig. 2 we show the raw judgements of humans and of our models, along with a

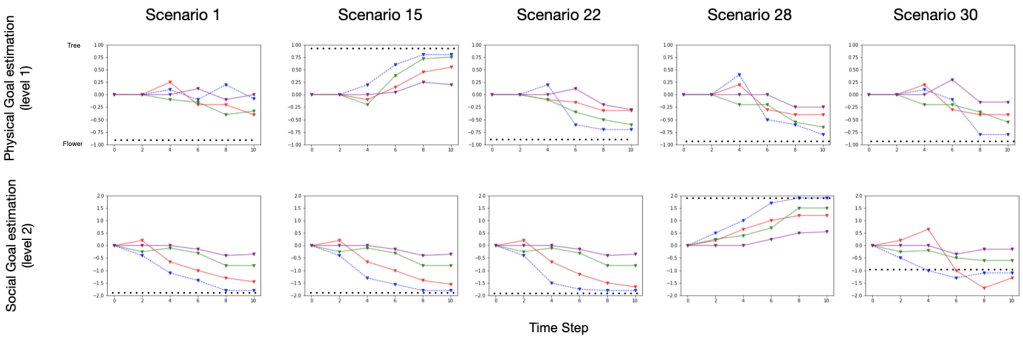


Figure 4: A deep dive into how humans and each model interprets all five scenarios at both levels at each time step (in red Human scores, in blue our Social MDP scores, in green Inverse planning [11], in purple the cue-based model [18], and **in dotted the ground truth**). The goal of each model is to interpret how one agent perceives another. (top) At level one, an agent has a belief over the physical goal of another agent. Humans and models predict what this belief is (the degree to which the agent believes that the other agent is heading toward the tree or the flower). Note that all models perform rather well and follow human judgements. (bottom) At level two, an agent has a belief over the physical and social goals of another agent. Humans and models predict what the beliefs of the agents are about the social goals of other agents. In other words, to what degree does this agent think that the other agent is hindering or helping them. Here our model fits human data much better because of its recursive nature. At deeper levels, our model is capable of capturing social interactions and social inferences that other models cannot. Other models are confused, and so predict that there is a very weak or non-existent social goal in most cases while our model follows human judgements.

best linear fit. The performance, by measuring correlation against human judgements, of all models is shown in fig. 3. Our model performs considerably better than other models. This is even more evident in the deep dive shown in fig. 4. For level one agents, agents that are social but that assume that other agents are not social, all models agreed with human judgements. Yet, for level two agents, agents that are social and can assume that other agents are also social, our models are far better aligned with human judgements. Prior work could capture a far smaller space of social interactions than the Social MDPs we craft here.

5 Conclusion

Social MDPs are a first step toward a theory of social interactions that fits within the established frameworks we have in robotics. They can perform zero-shot social recognition and planning for diverse situations. The fact that MDPs can be extended in a natural way that is also computationally tractable to account for many social interactions by nesting inference and allowing models to take arbitrary functions of the estimated rewards of other agents has not been noted before. Our experiments clearly show that Social MDPs are superior to prior models and account for more social interactions.

We have only begun to explore what Social MDPs can represent in this work. The environment we consider is very simple, yet, at the same time, more than enough to differentiate Social MDPs from other models. We have unrolled Social MDPs only two levels, what exists at deep levels is still unclear. It is likely that humans do not perform deeply-nested recursive reasoning to carry out social interactions, although, what the cutoff is, and if Social MDPs are close enough to a human's mental model to allow for measuring that cutoff is unknown.

We would like to see in the future that any MDP-based system can be augmented to be social by a straightforward extension with Social MDPs. Much like virtually any approach can be easily augmented to partially-observed environments using POMDPs. Social MDPs and POMDPs are compatible, exploring their combinations and the implications of partial observability for social interactions remains as future work.

References

[1] R. Bellman. A markovian decision process. *Journal of mathematics and mechanics*, 6(5): 679–684, 1957.

[2] K. J. Åström. Optimal control of markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1):174–205, 1965.

[3] P. J. Gmytrasiewicz and P. Doshi. Interactive pomdps: Properties and preliminary results. In *International Conference on Autonomous Agents: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-*, volume 3, pages 1374–1375, 2004.

[4] S. V. Albrecht and P. Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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 INST OF TECH CAMBRIDGE DEPT OF BRAIN AND COGNITIVE SCIENCES, 2009.

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

[13] A. Patron-Perez, M. Marszalek, I. Reid, and A. Zisserman. Structured learning of human interactions in tv shows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (12):2441–2453, 2012.

[14] M. J. Marin-Jimenez, A. Zisserman, M. Eichner, and V. Ferrari. Detecting people looking at each other in videos. *International Journal of Computer Vision*, 106(3):282–296, 2014.

[15] M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *2009 IEEE 12th international conference on computer vision*, pages 1593–1600. IEEE, 2009.

[16] X. Puig, T. Shu, S. Li, Z. Wang, J. B. Tenenbaum, S. Fidler, and A. Torralba. Watch-and-help: A challenge for social perception and human-ai collaboration. In *The International Conference on Learning Representations*, 2021.

[17] F. Heider and M. Simmel. An experimental study of apparent behavior. *The American journal of psychology*, 57(2):243–259, 1944.

[18] T. Shu, M. Kryven, T. D. Ullman, and J. B. Tenenbaum. Adventures in flatland: Perceiving social interactions under physical dynamics. In *42d proceedings of the annual meeting of the cognitive science society*, 2020.

[19] M. Kryven, T. Ullman, B. Cowan, and J. Tenenbaum. Plans or outcomes: How do we attribute intelligence to others? 2021.

[20] A. Netanyahu, T. Shu, B. Katz, A. Barbu, and J. B. Tenenbaum. Phase: Physically-grounded abstract social events for machine social perception. In *The AAAI Conference on Artificial Intelligence*, 2020.

[21] P. Doshi and P. J. Gmytrasiewicz. Monte carlo sampling methods for approximating interactive pomdps. *Journal of Artificial Intelligence Research*, 34:297–337, 2009.

[22] P. Doshi and D. Perez. Generalized point based value iteration for interactive pomdps. In *AAAI*, pages 63–68, 2008.

[23] B. Ng, K. Boakye, C. Meyers, and A. Wang. Bayes-adaptive interactive pomdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, 2012.

[24] 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.

[25] M. Igl, L. Zintgraf, T. A. Le, F. Wood, and S. Whiteson. Deep variational reinforcement learning for POMDPs. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2117–2126. PMLR, 10–15 Jul 2018.

[26] A. X. Lee, A. Nagabandi, P. Abbeel, and S. Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[27] C. F. Camerer, T.-H. Ho, and J.-K. Chong. A cognitive hierarchy model of games. *The Quarterly Journal of Economics*, 119(3):861–898, 2004.

[28] I. Shpitser, R. J. Evans, T. S. Richardson, and J. M. Robins. Introduction to nested markov models. *Behaviormetrika*, 41(1):3–39, 2014.

[29] W. Yoshida, R. J. Dolan, and K. J. Friston. Game theory of mind. *PLoS Comput Biol*, 4(12): e1000254, 2008.

[30] T. N. Hoang and K. H. Low. Interactive pomdp lite: Towards practical planning to predict and exploit intentions for interacting with self-interested agents. *arXiv preprint arXiv:1304.5159*, 2013.

[31] C. Baker, R. Saxe, and J. Tenenbaum. Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.