

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 ([Markov decision processes](#)) 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 ([Interactive-partially observed Markov decision processes](#)) 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 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. [While microsociology has uncovered common structures in social interactions \[1\] a computational model is still elusive.](#) Until not too long ago, this was also the state of robot navigation and sensing which was revolutionized by extending MDPs [2] to POMDPs [3]. 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 both physical goals ([e.g., bring the red key home](#)) and social goals ([e.g., prevent John from getting his yellow key](#)), and that their overall reward structure is an arbitrary combination of the two, potentially accompanied by other terms. Physical goals are precisely what MDPs can already express, a 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 a 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 a bounded depth, much like interactive POMDPs [4], I-POMDPs. Unlike I-POMDPs, Social MDPs are not recursive in terms of agent’s beliefs about the

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

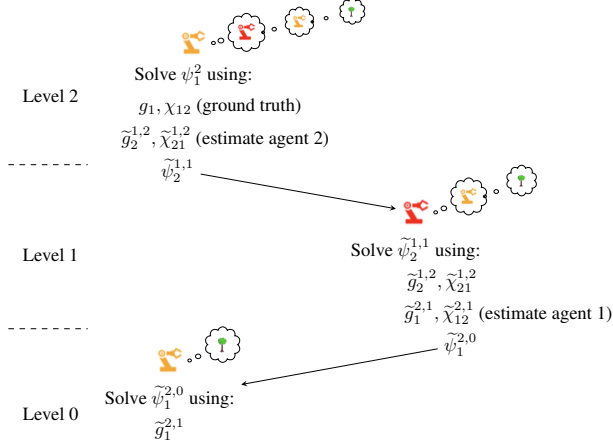


Figure 1: An example of recursively solving Social MDP for the yellow robot at level 2 in a two-agent interaction scenario. We denote the yellow robot as agent 1 and the red robot as agent 2. At level 2, the yellow robot estimates the red robot’s goals (both physical $\tilde{g}_2^{1,2}$ and social $\tilde{\chi}_{12}^{2,1}$) and social policy by assuming the red robot is running a level 1 Social MDP. Solving the social policy $\tilde{\psi}_2^{1,1}$ of the red robot at level 1 requires the red robot to estimate the yellow robot’s goals and policy by assuming the yellow robot is running a level 0 Social MDP, i.e., a regular MDP, so we can drop the estimation of $\tilde{\chi}_{12}^{2,1}$ here. All these estimates are in agent 1’s belief space and are updated at every time step.

```

Require:  $l, s^t, a_J^t, \chi_{ij}, g_i$ 
if  $l = 0$  then
  solve MDP for agent  $i$ 
else
  for all  $\tilde{\chi}_{ji}^{i,l,t}, \tilde{g}_j^{i,l,t}$  do
    compute
       $P(\tilde{\chi}_{ji}^{i,l,t} | s^{t-1}, a_J^{t-1})$ 
       $P(\tilde{g}_j^{i,l,t} | s^{1:t-1})$ 
       $\tilde{\psi}_j^{i,l-1}(s^t, a_J^t, \tilde{\chi}_{ji}^{i,l,t}, \tilde{g}_j^{i,l,t})$ 
    end for
    compute  $R_i^l(s^t, a_J^t, \chi_{iJ}, g_i)$ 
    compute  $Q_i^l(s^t, a_J^t, \chi_{iJ}, g_i)$ 
     $\pi_i^l \leftarrow \operatorname{argmax}_{a_i \in \mathcal{A}_i} Q_i^l$ 
  end if

```

Figure 2: The algorithm to compute social policy ψ_i^l for agent i at level l and time t . We use the estimated social policy $\tilde{\psi}_j^{i,l-1}$ at previous time step to update the estimated physical and social goal as described in section 3.3.1. At $t = 0$, we assume $P(\tilde{g}_j^{i,l,t})$ and $P(\tilde{\chi}_{ji}^{i,l,t})$ are from uniform distributions. This algorithm is called at all recursion steps $\tilde{\psi}_j^{i,l-1}$ to estimate social policy for the other agent $j \in J$. The estimated goals and policies are used to compute the rewards and Q values for selecting the actions.

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

53 Our contributions are: (1) formulating Social MDPs where an agent’s reward function is an arbitrary
 54 function of the recursive estimate of another agent’s reward and a physical goal, (2) an implementation
 55 where that function is a linear transformation, which captures many notions of helping and hindering,
 56 (3) demonstrating that the model performs zero-shot social reasoning in agreement with a human
 57 subjects experiment, and (4) examples of the practical utility of recursive social reasoning. In an
 58 anonymized online appendix¹ we fully enumerate all possible scenarios predicted by our model given
 59 an environment simple enough to allow doing so, demonstrating that it captures a diverse set of social
 60 behaviors. We also provide videos of the behavior of our model in all these scenarios.

61 2 Related Work

62 **Modeling other agents** In order to interact with other agents effectively, an agent must be able
 63 to reason about the goals, preferences, and beliefs of other agents [5]. Theory-based models for
 64 social goal attribution [6, 7, 8, 9, 10], Bayesian inverse planning to infer an agent’s goal given the
 65 observations of their behaviors [11, 12], and learning the reward functions of other agents [13] have
 66 been explored. Prior research also tried to recognize social interactions such as waving and hugging
 67 in videos where people are involved in group activities [14, 15, 16]. These methods generally involve
 68 two separate stages [17]: a social perception stage and a coordination or collaboration stage where
 69 agents interact. In contrast, Social MDPs constantly reevaluate the goals of other agents enabling
 70 them to adapt to changes in the plans of other agents. Social MDPs also allow for enumerating social
 71 situations by formally defining the space of what social interactions are, opening the doors to a more
 72 theoretical approach to social interactions. Game theory [18, 19] considers altruistic and spiteful
 73 behavior through linear combinations of payoffs, similarly to what we consider here although such

¹See <https://social-mdp.github.io>

74 models are limited to level 1 reasoning and to scenarios where the goals are known a prior rather than
75 estimated on the fly as they are in Social MDPs and in practical robotic applications.

76 **Simulating social interactions** Heider and Simmel [20] simulated agent trajectories to study social
77 perceptions through a set of animations involving the movements of geometrical figures. Simulating
78 agent behaviors in physics engines had been explored to collect datasets in fully observable [21, 22]
79 and partially observable environments [23, 17]. The collected datasets are used to study social
80 perception or build machine learning models that can recognize agents’ goals. These frameworks
81 assume that each agent has either a physical goal or a social goal. Here we consider a more general
82 scenario where agents have both a physical goal and a social goal and the overall behavior is a blend
83 of the two. This is a more practical setting as when one decides to help another agent one does not
84 abandon every other concern and physical care. [Our formulation allows for an arbitrary function of](#)
85 [another agent’s reward recognizing for complex social interactions.](#) We explore only a subset of the
86 full power of Social MDPs here by considering only linear functions.

87 **Learning to interact with other agents** Interactive POMDPs [24, 25, 26] (I-POMDPs) are exten-
88 sions of POMDPs that recursively model the beliefs of other agents. Social MDP and I-POMDPs are
89 orthogonal. Social MDPs allow agents to reason recursively about other agents’ reward functions
90 while I-POMDPs allow agents to reason recursively about other agent’s beliefs about the state of the
91 world. The two could in principle be combined, but while Social MDPs require solving a modest
92 number of additional nested MDPs, I-POMDPs require significantly nested inference, and when
93 the two are combined the problem quickly becomes intractable. [Xie et al. \[27\]](#) propose a different
94 type of approach that does not require nested inference: learning a low-dimensional representation
95 of another agent’s strategy. This approach allows an agent to avoid another agent or to manipulate
96 another agent into some mutually-beneficial behavior. Social MDPs, on the other hand, allow building
97 the strategy of another agent directly into the reward function of an agent, enabling behaviors such
98 as helping or hindering regardless of what the other agent is trying to achieve. Moreover, Social
99 MDPs are zero-shot, while this prior approach is not. From the point of view of generalization and
100 sample-efficient robotics, a zero-shot approach is preferable; in addition, it opens new doors for a
101 more theoretical understanding of social interactions. We could combine Social MDPs with this prior
102 work to build in latent representations of strategies into reward functions [28, 29] creating more
103 efficient approximations of Social MDPs.

104 3 Social MDPs

105 Social MDPs are recursive MDPs (Markov decision process) with nested estimates of other [agent’s](#)
106 [goals](#). They are inspired by hierarchical models of games [30] and nested MDP that reason about
107 the beliefs of other agents [31, 32, 33]. [Figure 1 shows an example of recursively estimating the](#)
108 [other agent’s goals and policy in a two-agent scenario.](#) Like other nested models, e.g I-POMDPs,
109 Social MDPs have the notion of a level. A level 0 Social MDP is simply an MDP: agents reason
110 about the map state. A level 1 Social MDP enables each agent to reason about the physical goals of
111 other agents (those other agents are treated like level 0 agents). A level 2 Social MDP enables each
112 agent to reason about the level 1 social goals of other agents. To perform this nested inference, agents
113 must have access to another agents’ physical and social goals. These goals are estimated by solving
114 Social MDPs recursively at every level.

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

122 3.1 Assumptions

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

128 In this paper, we consider only pairwise Social MDPs; indeed, all our experiments involve settings
 129 where only two agents are present. N-way Social MDPs that consider social interactions between
 130 multiple agents at the same time are extensions of pairwise MDP and scale linearly at every level.
 131 Computationally however, n-way Social MDPs, as with any such model, scale exponentially in n .

132 As with many multiagent MDP settings, the experiments performed here assume that agents are
 133 optimal planners. This is not an inherent limitation of Social MDPs, but relaxing this assumption
 134 makes estimating the other agents' goals and rewards far more difficult.

135 3.2 Formal definition of Social MDPs

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

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

138 where \mathcal{S} is a set of states in the environment where $s \in \mathcal{S}$; $\mathcal{A} = \mathcal{A}_J$ is the set of joint moves of all
 139 agents in J . a_i is an action for agent i ; T is the probability distribution of going from state $s \in \mathcal{S}$ to
 140 next state $s' \in \mathcal{S}$ given actions of all agents in J : $T(s' | s, a_J)$; χ_{iJ} is agent i 's social goal toward
 141 every other agent in J . For convenience, χ_{iJ} is a shorthand for $\bigcup_{j \in J, j \neq i} \chi_{ij}$; g_i is agent i 's physical
 142 goal; R_i^l is the l -th level reward function for agent i based on its estimate of other agents' rewards;
 143 and γ is a discount factor, $\gamma \in (0, 1)$.

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

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

148 where $r(\cdot)$ is the static reward given the agent's own physical goal g_i , $\tilde{R}_j^{i, l-1}(\cdot)$ is the estimated
 149 reward for agent j from agent i 's point of view assuming agent j is a level $l-1$ agent, $c(\cdot)$ is the cost
 150 for taking an action. For negative levels, the reward is defined to be zero.

151 χ_{ij} is the social goal, it transforms the reward of another agent j into a goal that is part of the reward
 152 of the target agent i . In this paper, we instantiate the model with a linear transformation, so χ_{ij}
 153 is simply a reweighting of the estimated reward of the other agent. If it is a negative value, the
 154 target agent will attempt to minimize the reward of another agent, i.e. hindering. A positive value
 155 corresponds to helping. Social goals can be eliminated entirely by setting this weight to zero.

156 In order to estimate another agent's reward function, one needs to estimate that agent's physical and
 157 social goals. We use $\tilde{\chi}_{jJ}^{i, l}$ and $\tilde{g}_j^{i, l}$ to denote the estimated social and physical goals. The superscript
 158 i, l indicates agent i at level l is making the estimations. We describe how to estimate the social and
 159 physical goals in section 3.3.1.

160 3.3 Planning for Social MDPs

161 Analogous to MDPs, the Q function of Social MDPs is the sum of immediate reward and the expected
 162 value in the future.

$$Q_i^l(s, a_J, \chi_{iJ}, g_i) = R(s, a_i, \chi_{iJ}, g_i) + \gamma \sum_{s' \in \mathcal{S}} T(s, a_J, s') V_i^l(s', \chi_{iJ}, g_i) \quad (3)$$

163 Since agent i is interacting with other agents $j \in J$, it needs to estimate what actions other agents are
 164 likely to take in order to compute its state-action value. Social MDPs take the expectation over the
 165 estimated goals and actions of agent j to compute $V_i^l(s', \chi_{iJ}, g_i)$:

$$\begin{aligned} V_i^l(s', \chi_{iJ}, g_i) &= \max_{a'_i \in \mathcal{A}_i} \left\{ E_{\tilde{g}_j^{i, l}, \tilde{\chi}_{jJ}^{i, l}, a'_j} [Q_i^l(s', a'_J, \chi_{iJ}, g_i)] \right\} \\ &= \max_{a'_i \in \mathcal{A}_i} \left\{ \sum_{\substack{j \in J \\ j \neq i}} \sum_{a'_j \in \mathcal{A}_j} \sum_{\tilde{g}_j^{i, l}} \int_{\tilde{\chi}_{jJ}^{i, l}} \underbrace{P(\tilde{g}_j^{i, l} | s^{1:t})}_{\text{estimate physical goal (Eq. 6)}} \underbrace{P(\tilde{\chi}_{jJ}^{i, l} | s, a_J)}_{\text{estimate social goal (Eq. 5)}} \underbrace{\tilde{\psi}_j^{i, l-1}(s', a'_J, \tilde{\chi}_{jJ}^{i, l}, \tilde{g}_j^{i, l})}_{\text{estimate social policy (Eq. 7)}} Q_i^l(\cdot) d\tilde{\chi}_{jJ}^{i, l} \right\} \quad (4) \end{aligned}$$

166

167 When solving agent i 's MDP at level l , the estimated social and physical goals are further used to
 168 update the other agent j 's social policy to the actions agent j may take. We denote the estimated
 169 social policy for agent j at reasoning level $l - 1$ as $\tilde{\psi}_j^{i,l-1} : \mathcal{S} \times \mathcal{A}_J \times \tilde{\chi}_{j,J}^{i,l} \times \tilde{g}_j^{i,l} \rightarrow [0, 1]$. Figure 2
 170 summarizes the steps to compute the state-action values and select optimal actions for any level l at
 171 time step t . We first update the probability of the estimated goals of other agents using the observed
 172 state and the estimated policy from the previous time step. The updated probability of goals are used
 173 to update the policy of other agents and compute the reward and Q function of the target agent. The
 174 recursion happens at estimating the social policies of other agent at a lower level.

175 3.3.1 Updating social and physical goals of other agents

176 An agent's estimate of another agent's social and physical goals at time step t and level l can
 177 be updated based on the actions performed by the agents. At time step $t = 0$, we use uniform
 178 distributions for social and physical goals.

179 The social goal, estimated at time step t , is updated after actions taken by all agents at the previous
 180 time step. This update is similar to the belief update in the POMDP framework but based on the
 181 estimated social policy of the other agent j :

$$P(\tilde{\chi}_{ji}^{i,l,t} | s^{t-1}, a_J^{t-1}) \propto P(\tilde{\chi}_{ji}^{i,l,t-1} | s^{t-2}, a_J^{t-2}) \sum_{\tilde{g}_j^{i,l,t-1}} P(a_j^{t-1} | s^{t-1}, \tilde{\chi}_{ji}^{i,l,t-1}, \tilde{g}_j^{i,l,t-1}) \times T(s^{t-1}, a_J^{t-1}, s^t) \quad (5)$$

182

183 The physical goal g_j of agent j is estimated by i as follows, similar to Shu et al. [21] but marginalized
 184 over the estimated social goal as the agent is estimating the social goal at the same time.

$$P(\tilde{g}_j^{i,l,t} | s^{1:t-1}) \propto \int_{\tilde{\chi}_{ji}^{i,l,t}} P(s^{1:t-1} | \tilde{g}_j^{i,l,t}, \tilde{\chi}_{ji}^{i,l,t}) \cdot P(\tilde{g}_j^{i,l,t}) \cdot P(\tilde{\chi}_{ji}^{i,l,t}) d\tilde{\chi}_{ji}^{i,l,t} \quad (6)$$

185 3.3.2 Estimating social policies of other agents

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

$$\tilde{\psi}_j^{i,l-1}(s, a_J, \tilde{\chi}_{j,J}^{i,l}, \tilde{g}_j^{i,l}) = \text{Softmax}(Q_j^{l-1}(s, a_J, \tilde{\chi}_{j,J}^{i,l}, \tilde{g}_j^{i,l})) \quad (7)$$

187 This is a softmax policy where we use a temperature parameter τ to control how much the agent j
 188 follows the greedy actions. As shown in eq. 4, in order to use agent j 's Q function at level $l-1$, it
 189 requires to compute agent i 's Q function at level $l-2$, and so on. This involves solving Social MDPs
 190 recursively at levels $0, 1, \dots, l-1$.

191 3.4 Time complexity

192 The time complexity of solving a Social MDP at level 0 is the same as that of solving an MDP. At
 193 level 1, an MDP must be solved for every agent independently in order to compute the likely physical
 194 goals of every other agent. Assume that the number of models considered for each pair of agents at
 195 each level is bounded by a number M (based on the number of social and physical goals to consider).
 196 Solving a Social MDP at level l requires solving $O(M(A-1)^2l)$ MDPs, where A is the number
 197 of agents. Social MDPs form a tree with branching factor $A - 1$ as every agent must compute the
 198 pairwise social goal of every other agent until level 0 where the tree bottoms out. There are many
 199 potential speedups that can alleviate this runtime to allow for efficient inference even in the face of
 200 many agents. For example, a distance horizon could be used where far away agents could simply
 201 be considered non-interacting. Similar to Netanyahu et al. [23], it is also possible to speed up the
 202 algorithm by amortized inference over goals and relations by training a neural net to recognize goals
 203 and relations as initial guesses and refine them through probabilistic inference.

204 4 Results

205 We apply the Social MDP framework to a multi-agent grid world inspired by previous studies on
 206 social perceptions [12, 6, 34]. The 10×10 world consists of two agents, a yellow robot and red robot,
 207 two physical landmarks, a flower and tree, and two objects, a yellow watering can and red watering
 208 can. The yellow agent has a low cost for moving the yellow watering can, while it has a high cost for
 209 moving the red watering can. Robots can have a physical goal of moving the watering can to a target

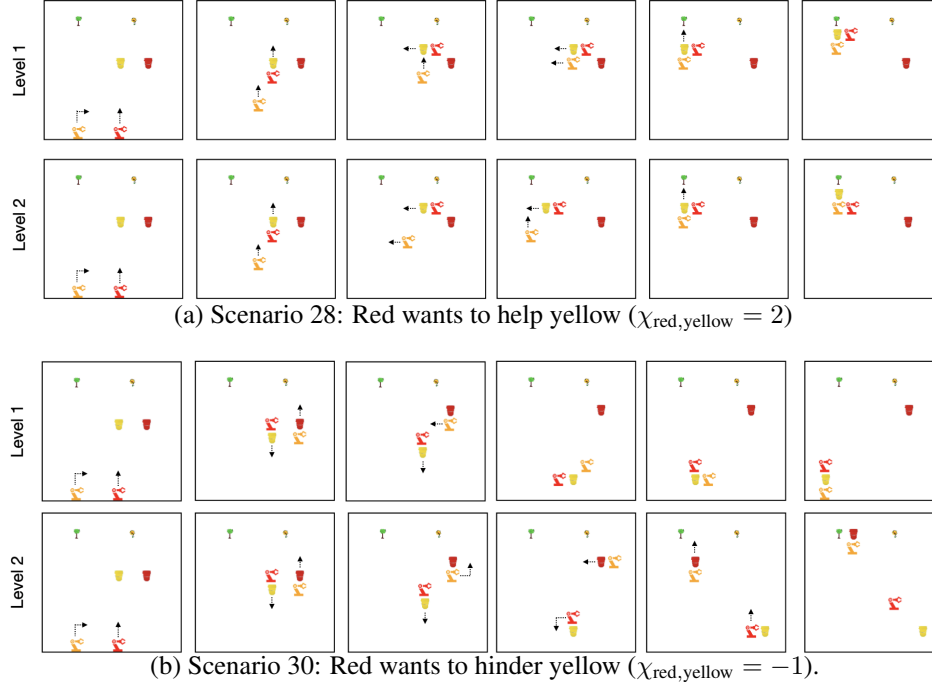


Figure 3: Two examples of zero-shot social interactions. The Social MDP gives the robots the ability to understand and predict relationships, thereby making far more efficient actions. The yellow robot wants to water the tree. Moving the yellow watering can is easy for the yellow robot, while moving the red can 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 robot by moving its watering can toward the tree. At level 2, the yellow robot recognizes that red is helping, then estimates 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 hinder it. As such, it attempts to move the yellow can and repeatedly fails, entering a local minima where the yellow can is the one easiest to move without realizing that the red robot will forever prevent this. At level 2, the yellow robot recognizes that the red robot is attempting to hinder it, gives up on the yellow can, and makes a globally-optimal move of using the harder-to-move red can instead.

plant. Robots can have a social goal of helping or hindering to different degrees. In the grid world, agents can move in four directions (left, right, up, down) or choose not to move.

98 different experiment scenarios² are systematically created in this grid world. Each scenario has agents as having either the same physical 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 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 feature missing from alternative representations. All 98 experiment 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

²Interactions for the experimental scenarios can be viewed at <https://social-mdp.github.io/scenarios>

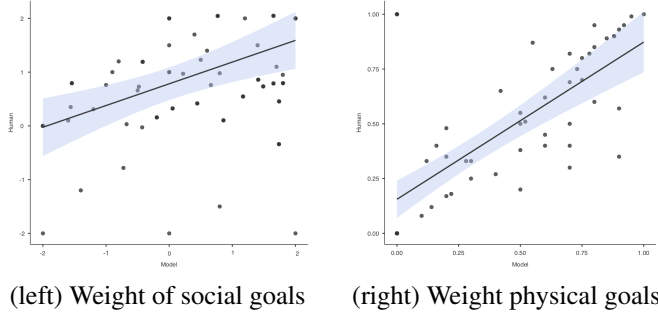


Figure 4: Twelve human subjects, and our model, the Social MDP, watched and scored 10 videos at different snapshots. These videos consist of five scenarios where robots reason at either level 1 or level 2 (presented to the users in randomized order). The straight black line represents the best linear fit to the data, and the light blue band around the line shows the uncertainty in the linear fit. The light blue band represents a 95% confidence interval. (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.

	<i>Social MDP (ours)</i>	Inverse Planning	Cue-based
Social Goal	0.85	0.78	0.20
Physical Goal	0.76	0.71	0.07

Table 1: The coefficient of correlation with 95% confidence interval between machine judgements and ground truth (final 2 time steps) for all the 98 experiment scenarios (each scenario has agents having either the same or different physical goals along with one of 7 different scaling factors on each of their social goals (-2, -1, -0.5, 0, 0.5, 1, 2)). Refer to Appendix for detailed results for each scenario. We provide two baselines and our own approach. The cue-based model is described in Shu et al. [21]. The inverse planning model is described in Ullman et al. [12]. Social MDPs produce better alignment with ground truth than other models and do not require training like the cue-based model.

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 [12]. This physical reward function is parameterized by ρ and δ that determines the scale and shape of the physical reward: $r_i(s, a, g_i) = \max(\rho(1 - \text{distance}(s, a, g_i)/\delta), 0)$. 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. 3. 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 [12] and a recent cue-based model [21]. 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. 4 we show the raw judgements of humans and of our models, along with a best linear fit. The performance of all models against human judgements, was measured through correlation coefficient at 95% confidence level, for social goal estimation ($r = 0.89$ for the Social MDP vs. $r = 0.81$ for the Inverse Planning model vs. $r = 0.23$ for the Cue-based model) and physical

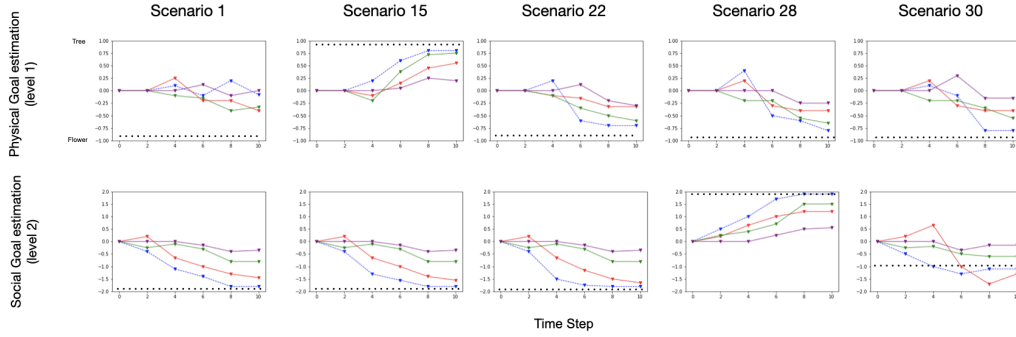


Figure 5: A deep dive into how humans and each model interpret the five experiment scenarios (refer to Appendix for results for all experiment scenarios) at both levels at each time step (in red Human scores, in blue our Social MDP scores, in green Inverse planning [12], in purple the cue-based model [21], 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.

248 goal estimation ($r = 0.78$ for the Social MDP vs. $r = 0.72$ for the Inverse Planning model vs. $r = 0.08$
 249 for the Cue-based model). Our model performs considerably better than other models. This is even
 250 more evident in the deep dive shown in fig. 5. For level one agents, agents that are social but that
 251 assume that other agents are not social, all models agreed with human judgements. Yet, for level
 252 two agents, agents that are social and can assume that other agents are also social, our models are
 253 far better aligned with human judgements. Prior work could capture a far smaller space of social
 254 interactions than the Social MDPs we craft here.

255 5 Conclusion

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

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

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

References

- [1] T. J. Scheff. *Microsociology: Discourse, emotion, and social structure*. University of Chicago Press, 1990.
- [2] R. Bellman. A markovian decision process. *Journal of mathematics and mechanics*, 6(5): 679–684, 1957.
- [3] K. J. Åström. Optimal control of markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1):174–205, 1965.
- [4] 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.
- [5] S. V. Albrecht and P. Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] C. L. Baker, R. Saxe, and J. B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.
- [12] 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.
- [13] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan. Cooperative inverse reinforcement learning. *Advances in neural information processing systems*, 29:3909–3917, 2016.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] D. K. Levine. Modeling altruism and spitefulness in experiments. *Review of economic dynamics*, 1(3):593–622, 1998.

- [19] Z. Feng, C. Tan, J. Zhang, and Q. Zeng. Bargaining game with altruistic and spiteful preferences. *Group Decision and Negotiation*, 30(2):277–300, 2021.
- [20] F. Heider and M. Simmel. An experimental study of apparent behavior. *The American journal of psychology*, 57(2):243–259, 1944.
- [21] 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.
- [22] M. Kryven, T. Ullman, B. Cowan, and J. Tenenbaum. Plans or outcomes: How do we attribute intelligence to others? In *PsyArXiv*, 2021.
- [23] 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.
- [24] P. Doshi and P. J. Gmytrasiewicz. Monte carlo sampling methods for approximating interactive pomdps. *Journal of Artificial Intelligence Research*, 34:297–337, 2009.
- [25] P. Doshi and D. Perez. Generalized point based value iteration for interactive pomdps. In *AAAI*, pages 63–68, 2008.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] I. Shpitser, R. J. Evans, T. S. Richardson, and J. M. Robins. Introduction to nested markov models. *Behaviormetrika*, 41(1):3–39, 2014.
- [32] W. Yoshida, R. J. Dolan, and K. J. Friston. Game theory of mind. *PLoS Comput Biol*, 4(12):e1000254, 2008.
- [33] T. N. Hoang and K. H. Low. Interactive pomdp lite: Towards practical planning to predict and exploit intentions for interacting with self-interested agents. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.
- [34] 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.

A Experiments

We list out all the possible experiment scenarios³ varied across physical goals and social goals for the two agents (yellow robot and red robot). Each scenario has agents as having either the same physical 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 have the factor set to zero; these agents only have a physical goal and no social goal.

Table 2: An enumeration of all 98 scenarios for different configurations of physical and social goals of the robots.

Scenario Id	Yellow Robot		Red Robot	
	Physical Goal	Social Scaling	Physical Goal	Social Scaling
S1	Tree	-2	Flower	-2
S2	Tree	-2	Flower	-1
S3	Tree	-2	Flower	-0.5
S4	Tree	-2	Flower	0
S5	Tree	-2	Flower	0.5
S6	Tree	-2	Flower	1
S7	Tree	-2	Flower	2
S8	Tree	-1	Flower	-2
S9	Tree	-1	Flower	-1
S10	Tree	-1	Flower	-0.5
S11	Tree	-1	Flower	0
S12	Tree	-1	Flower	0.5
S13	Tree	-1	Flower	1
S14	Tree	-1	Flower	2
S15	Flower	-0.5	Tree	-2
S16	Tree	-0.5	Flower	-1
S17	Tree	-0.5	Flower	-0.5
S18	Tree	-0.5	Flower	0
S19	Tree	-0.5	Flower	0.5
S20	Tree	-0.5	Flower	1
S21	Tree	-0.5	Flower	2
S22	Tree	0	Flower	-2
S23	Tree	0	Flower	-1
S24	Tree	0	Flower	-0.5
S25	Tree	0	Flower	0
S26	Tree	0	Flower	0.5
S27	Tree	0	Flower	1
S28	Tree	0	Flower	2
S29	Tree	0.5	Flower	-2
S30	Tree	0.5	Flower	-1
S31	Tree	0.5	Flower	-0.5
S32	Tree	0.5	Flower	0
S33	Tree	0.5	Flower	0.5
S34	Tree	0.5	Flower	1
S35	Tree	0.5	Flower	2
S36	Tree	1	Flower	-2
S37	Tree	1	Flower	-1

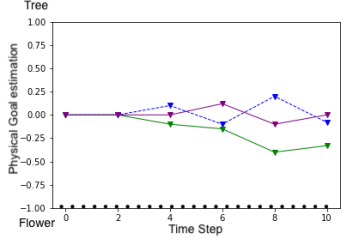
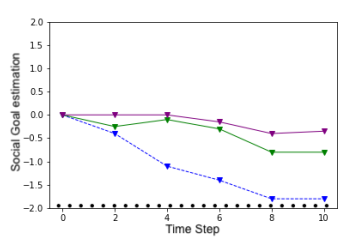
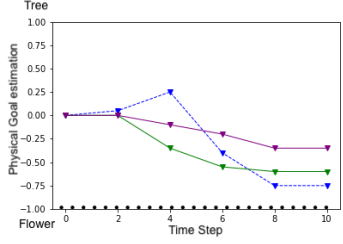
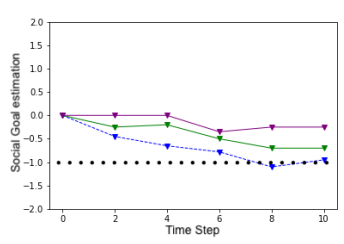
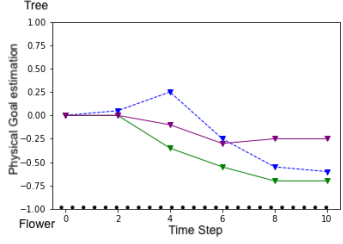
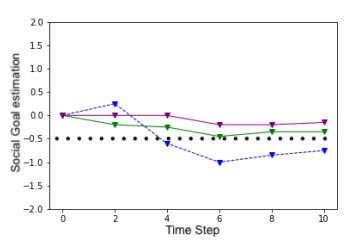
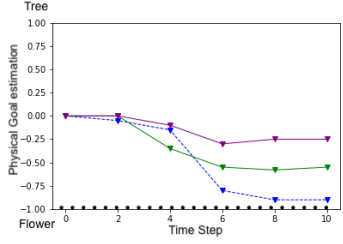
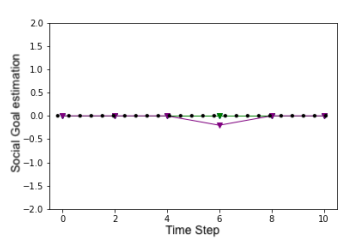
³Interactions for the experimental scenarios can be viewed at <https://social-mdp.github.io/scenarios>

S38	Tree	1	Flower	-0.5
S39	Tree	1	Flower	0
S40	Tree	1	Flower	0.5
S41	Tree	1	Flower	1
S42	Tree	1	Flower	2
S43	Tree	2	Flower	-2
S44	Tree	2	Flower	-1
S45	Tree	2	Flower	-0.5
S46	Tree	2	Flower	0
S47	Tree	2	Flower	0.5
S48	Tree	2	Flower	1
S49	Tree	2	Flower	2
S50	Tree	-2	Tree	-2
S51	Tree	-2	Tree	-1
S52	Tree	-2	Tree	-0.5
S53	Tree	-2	Tree	0
S54	Tree	-2	Tree	0.5
S55	Tree	-2	Tree	1
S56	Tree	-2	Tree	2
S57	Tree	-1	Tree	-2
S58	Tree	-1	Tree	-1
S59	Tree	-1	Tree	-0.5
S60	Tree	-1	Tree	0
S61	Tree	-1	Tree	0.5
S62	Tree	-1	Tree	1
S63	Tree	-1	Tree	2
S64	Tree	-0.5	Tree	-2
S65	Tree	-0.5	Tree	-1
S66	Tree	-0.5	Tree	-0.5
S67	Tree	-0.5	Tree	0
S68	Tree	-0.5	Tree	0.5
S69	Tree	-0.5	Tree	1
S70	Tree	-0.5	Tree	2
S71	Tree	0	Tree	-2
S72	Tree	0	Tree	-1
S73	Tree	0	Tree	-0.5
S74	Tree	0	Tree	0
S75	Tree	0	Tree	0.5
S76	Tree	0	Tree	1
S77	Tree	0	Tree	2
S78	Tree	0.5	Tree	-2
S79	Tree	0.5	Tree	-1
S80	Tree	0.5	Tree	-0.5
S81	Tree	0.5	Tree	0
S82	Tree	0.5	Tree	0.5
S83	Tree	0.5	Tree	1
S84	Tree	0.5	Tree	2
S85	Tree	1	Tree	-2
S86	Tree	1	Tree	-1
S87	Tree	1	Tree	-0.5
S88	Tree	1	Tree	0
S89	Tree	1	Tree	0.5
S90	Tree	1	Tree	1
S91	Tree	1	Tree	2
S92	Tree	2	Tree	-2
S93	Tree	2	Tree	-1
S94	Tree	2	Tree	-0.5
S95	Tree	2	Tree	0
S96	Tree	2	Tree	0.5

S97	Tree	2	Tree	1
S98	Tree	2	Tree	2

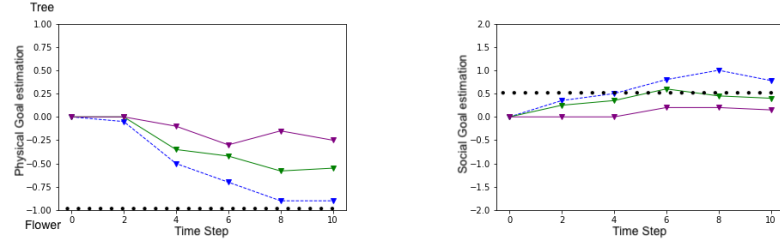
For each of the experiment scenario we show the Yellow robot's estimation of the physical and social goal of the red robot at each time step. The overall performance of S-MDP and the baselines against the ground truth is reported in table 1.

Table 3: Using S-MDP, at different levels of reasoning, Yellow robot estimates the physical and social goal of the red robot at each time step. The lines in blue is our Social MDP scores, in green is Inverse planning [12], in purple is the cue-based model [21], and in dotted is the ground truth).

Scenario Id	Physical goal estimation (Level 1)	Social goal estimation (Level 2)
S1	Red robot is initialized with the physical goal of Flower and social goal of -2. Yellow robot is initialized with the physical goal of Tree and social goal of -2.	
		
S2	Red robot is initialized with the physical goal of Flower and social goal of -1. Yellow robot is initialized with the physical goal of Tree and social goal of -2.	
		
S3	Red robot is initialized with the physical goal of Flower and social goal of -0.5. Yellow robot is initialized with the physical goal of Tree and social goal of -2.	
		
S4	Red robot is initialized with the physical goal of Flower and social goal of 0. Yellow robot is initialized with the physical goal of Tree and social goal of -2.	
		

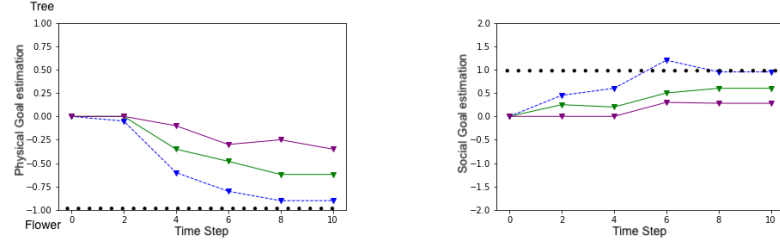
S5

Red robot is initialized with the physical goal of Flower and social goal of 0.5.
Yellow robot is initialized with the physical goal of Tree and social goal of -2.



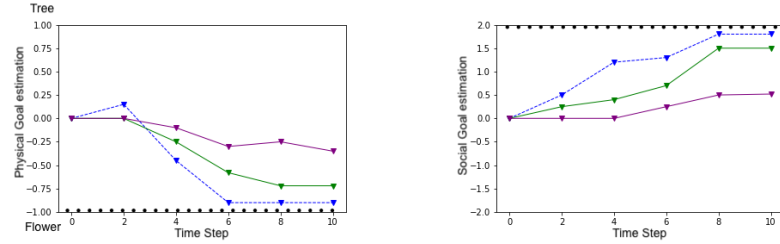
S6

Red robot is initialized with the physical goal of Flower and social goal of 1.
Yellow robot is initialized with the physical goal of Tree and social goal of -2.



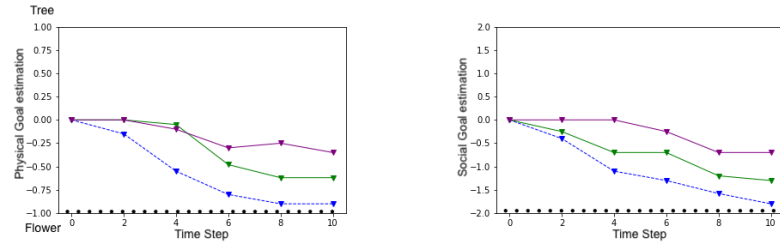
S7

Red robot is initialized with the physical goal of Flower and social goal of 2.
Yellow robot is initialized with the physical goal of Tree and social goal of -2.



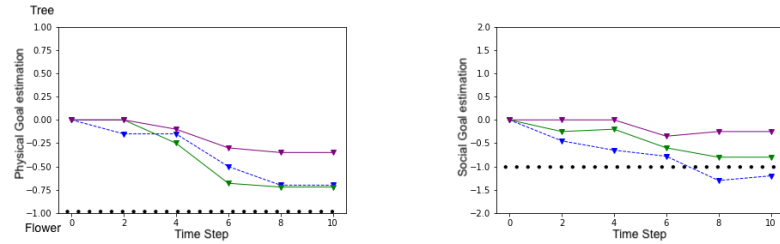
S8

Red robot is initialized with the physical goal of Flower and social goal of -2.
Yellow robot is initialized with the physical goal of Tree and social goal of -1.



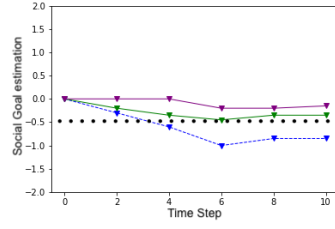
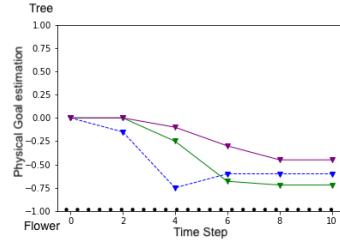
S9

Red robot is initialized with the physical goal of Flower and social goal of -1.
Yellow robot is initialized with the physical goal of Tree and social goal of -1.



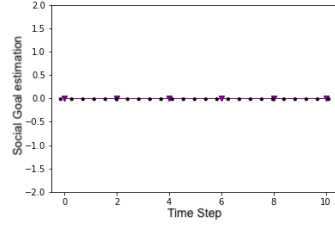
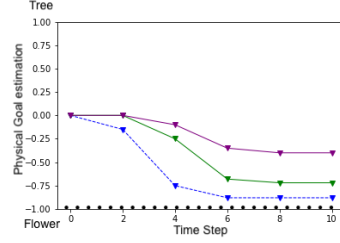
S10

Red robot is initialized with the physical goal of Flower and social goal of -0.5.
Yellow robot is initialized with the physical goal of Tree and social goal of -1.



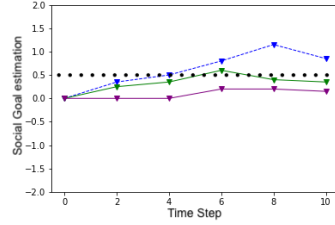
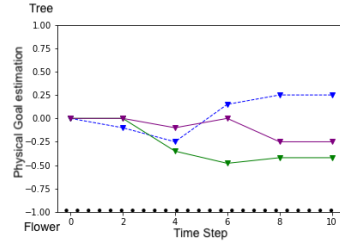
S11

Red robot is initialized with the physical goal of Flower and social goal of 0. Yellow robot is initialized with the physical goal of Tree and social goal of -1.



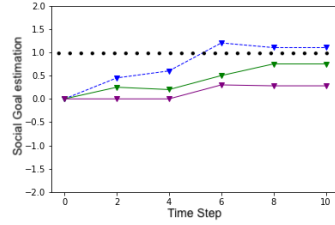
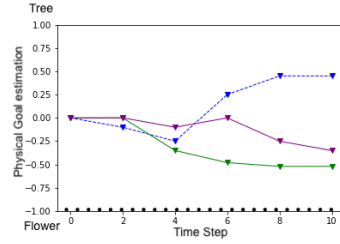
S12

Red robot is initialized with the physical goal of Flower and social goal of 0.5. Yellow robot is initialized with the physical goal of Tree and social goal of -1.



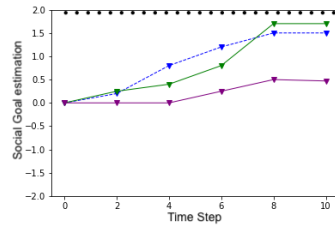
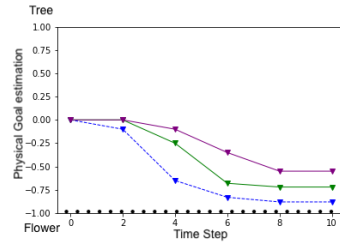
S13

Red robot is initialized with the physical goal of Flower and social goal of 1. Yellow robot is initialized with the physical goal of Tree and social goal of -1.



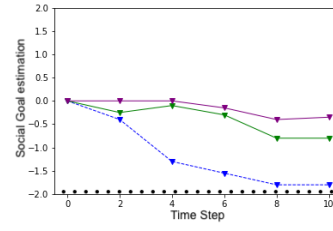
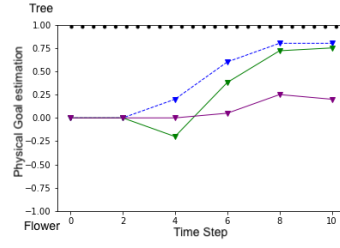
S14

Red robot is initialized with the physical goal of Flower and social goal of 2. Yellow robot is initialized with the physical goal of Tree and social goal of -1.



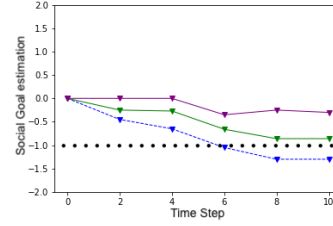
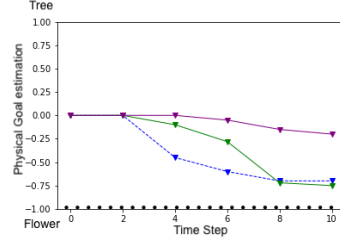
S15

Red robot is initialized with the physical goal of Tree and social goal of -2. Yellow robot is initialized with the physical goal of Flower and social goal of -0.5.



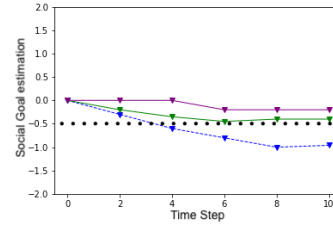
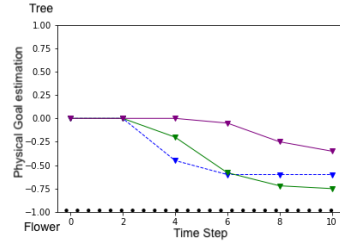
S16

Red robot is initialized with the physical goal of Flower and social goal of -1. Yellow robot is initialized with the physical goal of Tree and social goal of -0.5.



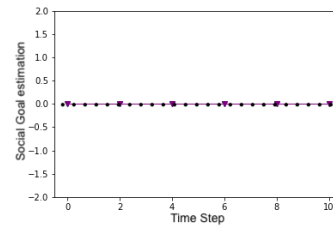
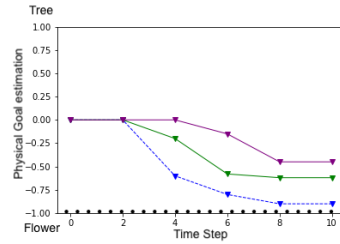
S17

Red robot is initialized with the physical goal of Flower and social goal of -0.5. Yellow robot is initialized with the physical goal of Tree and social goal of -0.5.



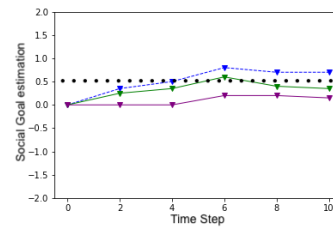
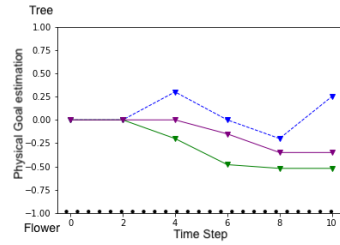
S18

Red robot is initialized with the physical goal of Flower and social goal of 0. Yellow robot is initialized with the physical goal of Tree and social goal of -0.5.



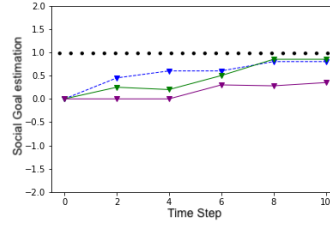
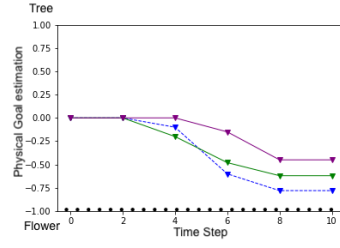
S19

Red robot is initialized with the physical goal of Flower and social goal of 0.5. Yellow robot is initialized with the physical goal of Tree and social goal of -0.5.



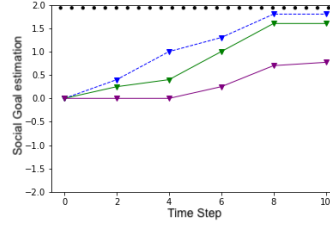
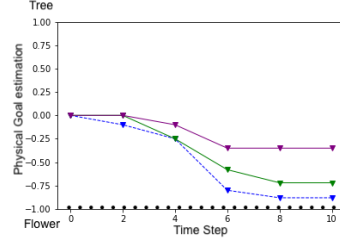
S20

Red robot is initialized with the physical goal of Flower and social goal of 1. Yellow robot is initialized with the physical goal of Tree and social goal of -0.5.



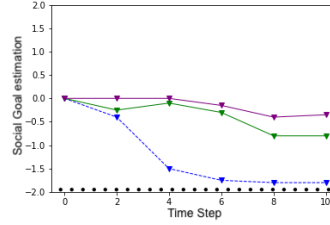
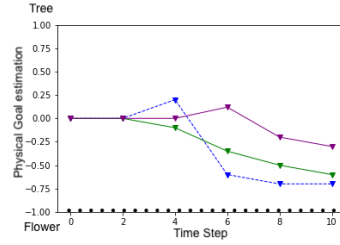
S21

Red robot is initialized with the physical goal of Flower and social goal of 2. Yellow robot is initialized with the physical goal of Tree and social goal of -0.5.



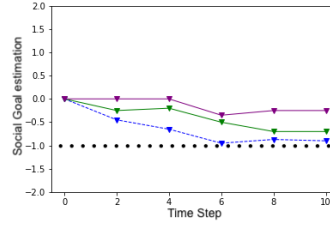
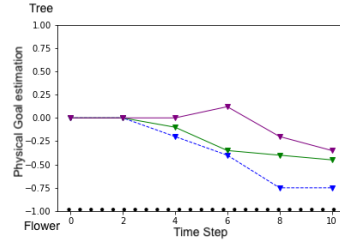
S22

Red robot is initialized with the physical goal of Flower and social goal of -2. Yellow robot is initialized with the physical goal of Tree and social goal of 0.



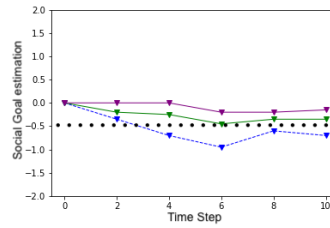
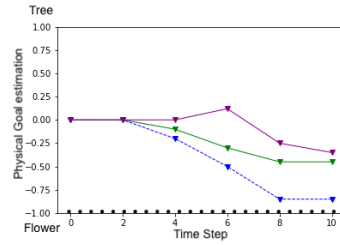
S23

Red robot is initialized with the physical goal of Flower and social goal of -1. Yellow robot is initialized with the physical goal of Tree and social goal of 0.



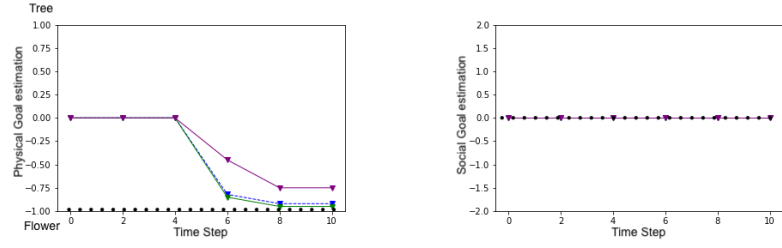
S24

Red robot is initialized with the physical goal of Flower and social goal of -0.5. Yellow robot is initialized with the physical goal of Tree and social goal of 0.

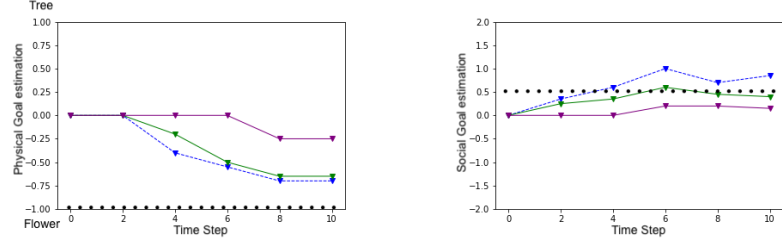


S25

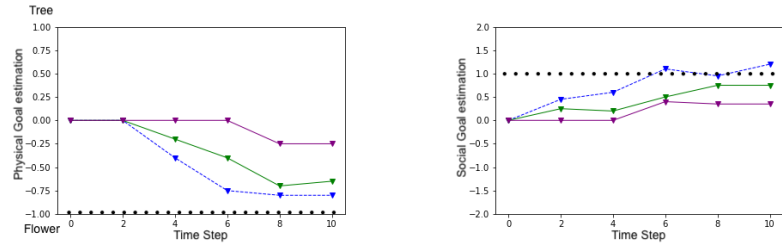
Red robot is initialized with the physical goal of Flower and social goal of 0. Yellow robot is initialized with the physical goal of Tree and social goal of 0.



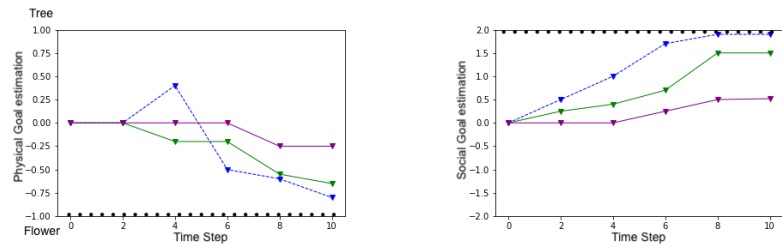
S26 Red robot is initialized with the physical goal of Flower and social goal of 0.5. Yellow robot is initialized with the physical goal of Tree and social goal of 0.



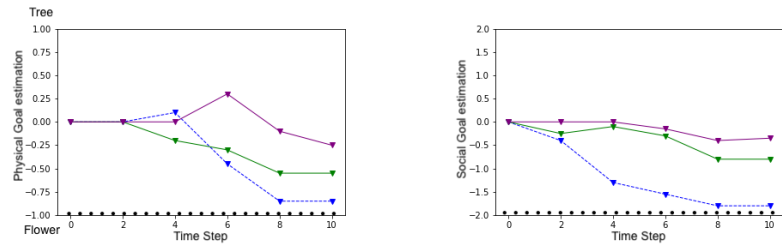
S27 Red robot is initialized with the physical goal of Flower and social goal of 1. Yellow robot is initialized with the physical goal of Tree and social goal of 0.



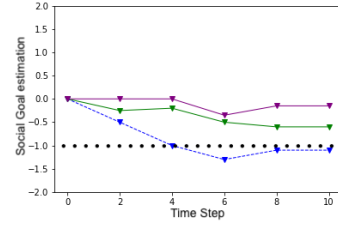
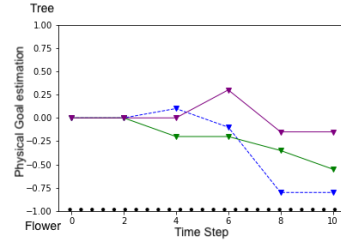
S28 Red robot is initialized with the physical goal of Flower and social goal of 2. Yellow robot is initialized with the physical goal of Tree and social goal of 0.



S29 Red robot is initialized with the physical goal of Flower and social goal of -2. Yellow robot is initialized with the physical goal of Tree and social goal of 0.5.

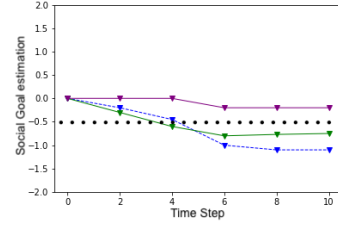
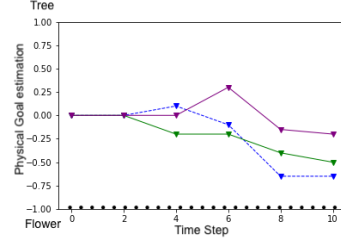


S30 Red robot is initialized with the physical goal of Flower and social goal of -1. Yellow robot is initialized with the physical goal of Tree and social goal of 0.5.



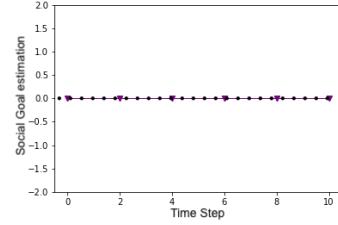
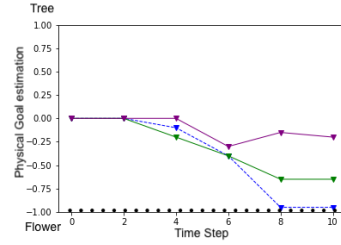
S31

Red robot is initialized with the physical goal of Flower and social goal of -0.5. Yellow robot is initialized with the physical goal of Tree and social goal of 0.5.



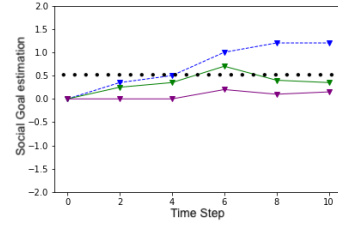
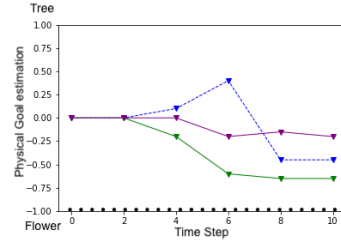
S32

Red robot is initialized with the physical goal of Flower and social goal of 0. Yellow robot is initialized with the physical goal of Tree and social goal of 0.5.



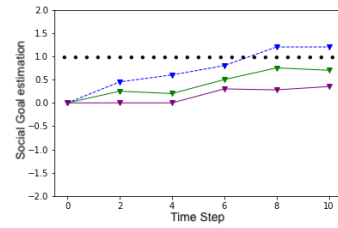
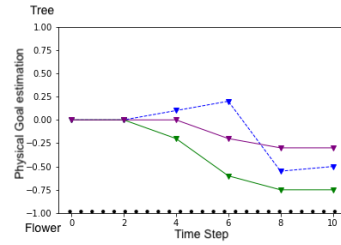
S33

Red robot is initialized with the physical goal of Flower and social goal of 0.5. Yellow robot is initialized with the physical goal of Tree and social goal of 0.5.



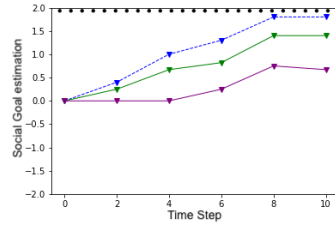
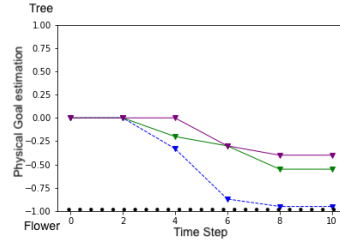
S34

Red robot is initialized with the physical goal of Flower and social goal of 1. Yellow robot is initialized with the physical goal of Tree and social goal of 0.5.



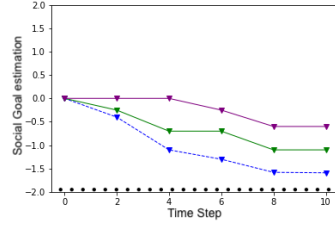
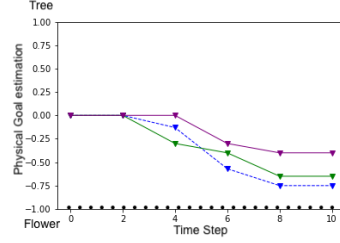
S35

Red robot is initialized with the physical goal of Flower and social goal of 2. Yellow robot is initialized with the physical goal of Tree and social goal of 0.5.



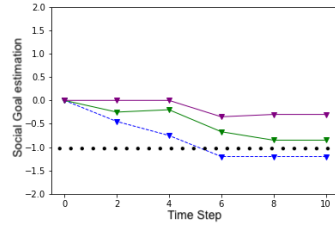
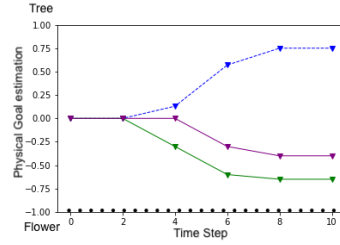
S36

Red robot is initialized with the physical goal of Flower and social goal of -2. Yellow robot is initialized with the physical goal of Tree and social goal of 1.



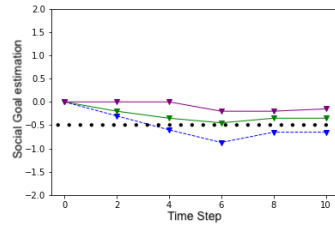
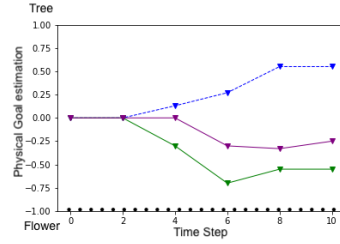
S37

Red robot is initialized with the physical goal of Flower and social goal of -1. Yellow robot is initialized with the physical goal of Tree and social goal of 1.



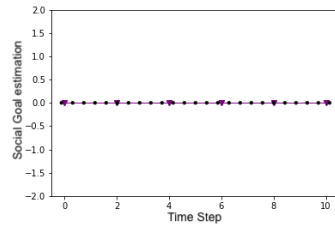
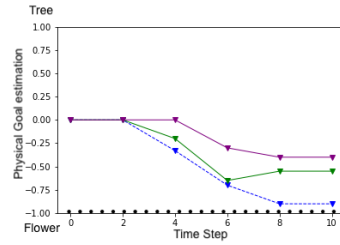
S38

Red robot is initialized with the physical goal of Flower and social goal of -0.5. Yellow robot is initialized with the physical goal of Tree and social goal of 1.



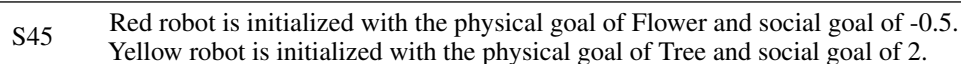
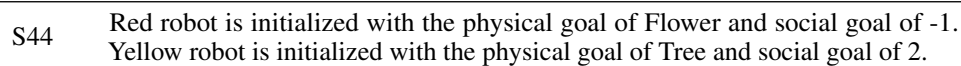
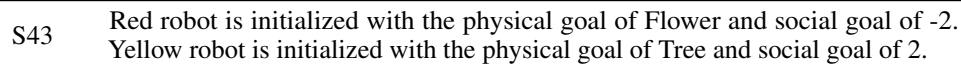
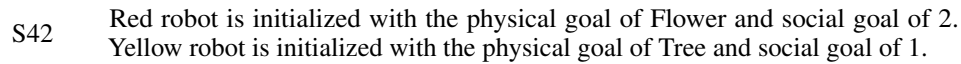
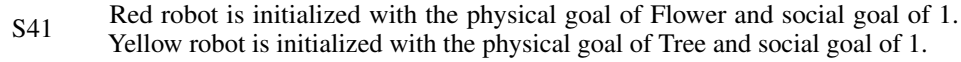
S39

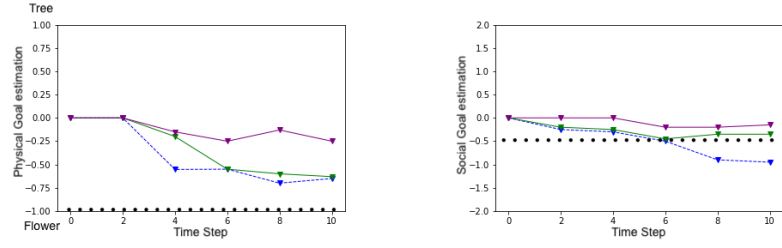
Red robot is initialized with the physical goal of Flower and social goal of 0. Yellow robot is initialized with the physical goal of Tree and social goal of 1.



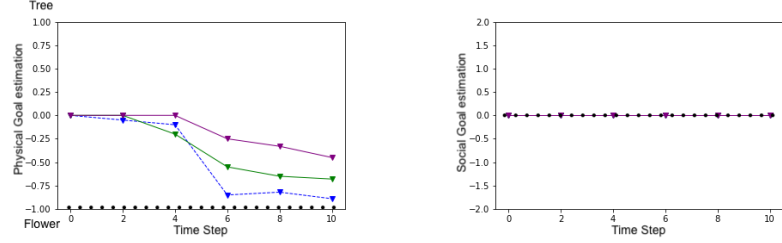
S40

Red robot is initialized with the physical goal of Flower and social goal of 0.5. Yellow robot is initialized with the physical goal of Tree and social goal of 1.

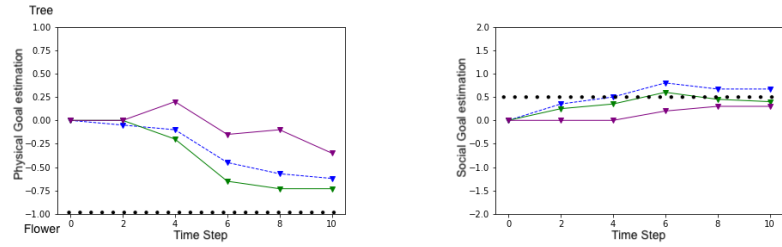




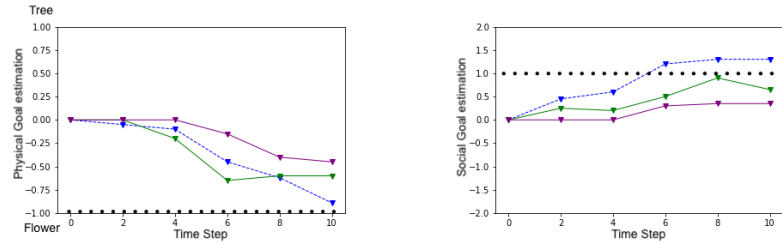
- S46 Red robot is initialized with the physical goal of Flower and social goal of 0. Yellow robot is initialized with the physical goal of Tree and social goal of 2.



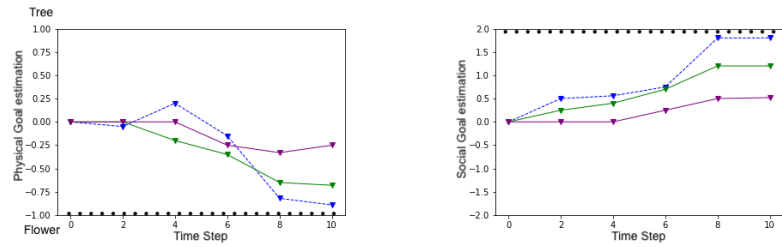
- S47 Red robot is initialized with the physical goal of Flower and social goal of 0.5. Yellow robot is initialized with the physical goal of Tree and social goal of 2.



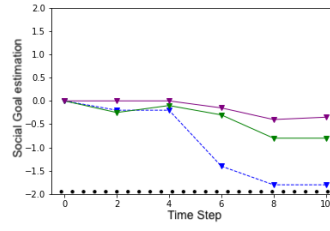
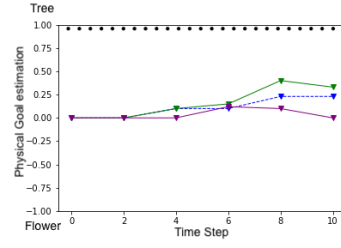
- S48 Red robot is initialized with the physical goal of Flower and social goal of 1. Yellow robot is initialized with the physical goal of Tree and social goal of 2.



- S49 Red robot is initialized with the physical goal of Flower and social goal of 2. Yellow robot is initialized with the physical goal of Tree and social goal of 2.

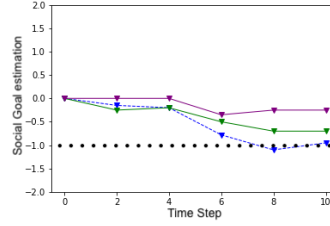
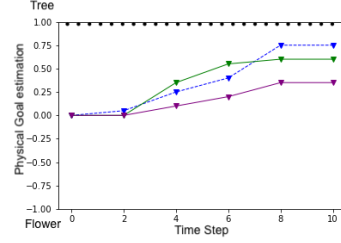


- S50 Red robot is initialized with the physical goal of Tree and social goal of -2. Yellow robot is initialized with the physical goal of Tree and social goal of -2.



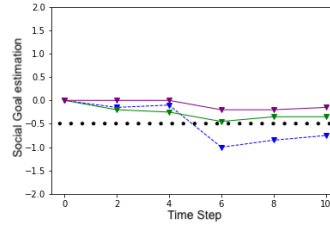
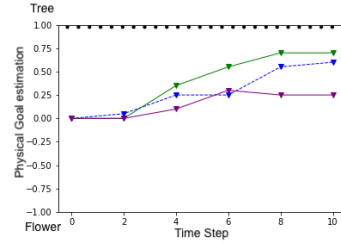
S51

Red robot is initialized with the physical goal of Tree and social goal of -1. Yellow robot is initialized with the physical goal of Tree and social goal of -2.



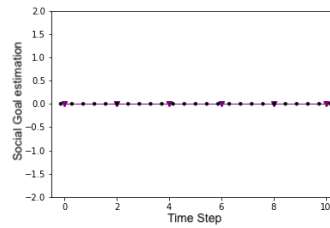
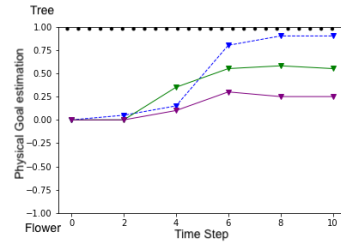
S52

Red robot is initialized with the physical goal of Tree and social goal of -0.5. Yellow robot is initialized with the physical goal of Tree and social goal of -2.



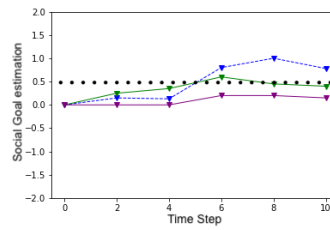
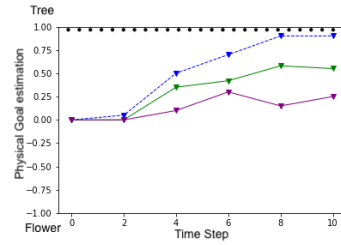
S53

Red robot is initialized with the physical goal of Tree and social goal of 0. Yellow robot is initialized with the physical goal of Tree and social goal of -2.



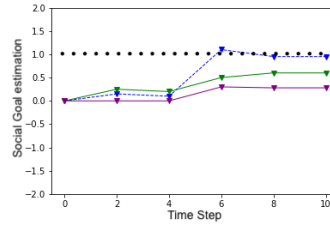
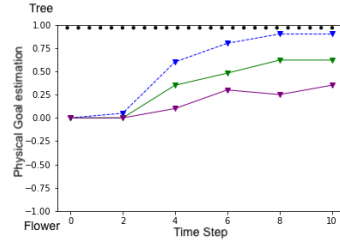
S54

Red robot is initialized with the physical goal of Tree and social goal of 0.5. Yellow robot is initialized with the physical goal of Tree and social goal of -2.



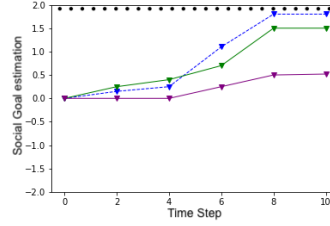
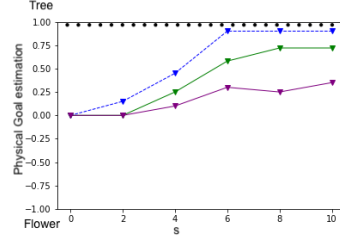
S55

Red robot is initialized with the physical goal of Tree and social goal of 1. Yellow robot is initialized with the physical goal of Tree and social goal of -2.



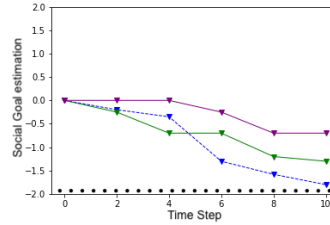
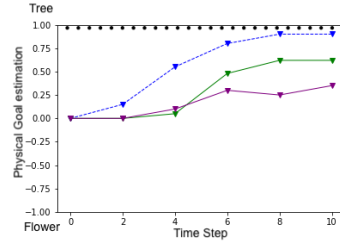
S56

Red robot is initialized with the physical goal of Tree and social goal of 2. Yellow robot is initialized with the physical goal of Tree and social goal of -2.



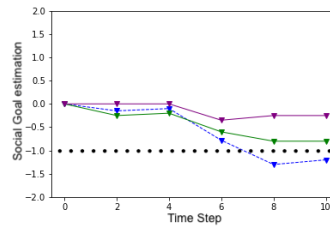
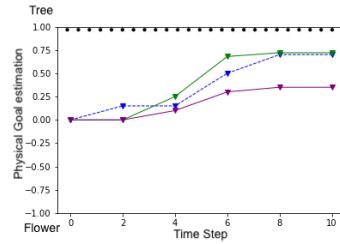
S57

Red robot is initialized with the physical goal of Tree and social goal of -2. Yellow robot is initialized with the physical goal of Tree and social goal of -1.



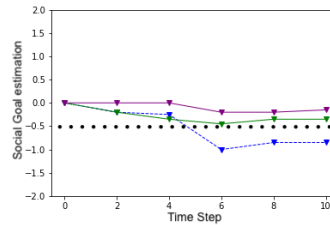
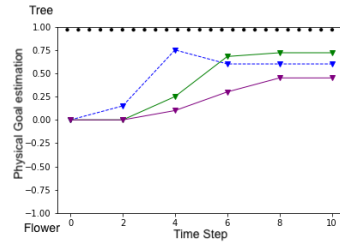
S58

Red robot is initialized with the physical goal of Tree and social goal of -1. Yellow robot is initialized with the physical goal of Tree and social goal of -1.



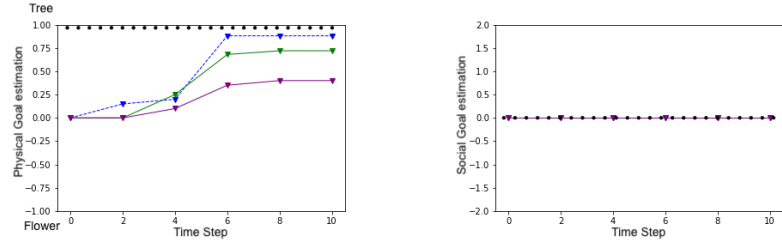
S59

Red robot is initialized with the physical goal of Tree and social goal of -0.5. Yellow robot is initialized with the physical goal of Tree and social goal of -1.

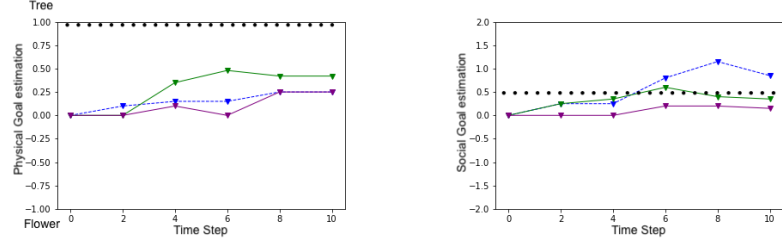


S60

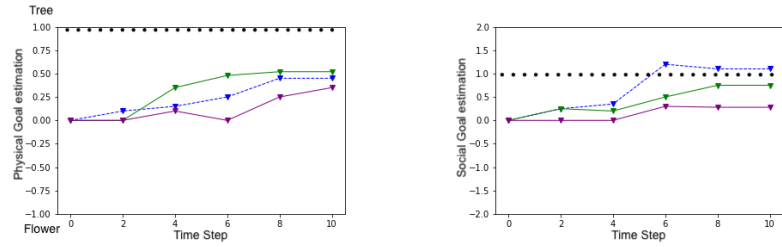
Red robot is initialized with the physical goal of Tree and social goal of 0. Yellow robot is initialized with the physical goal of Tree and social goal of -1.



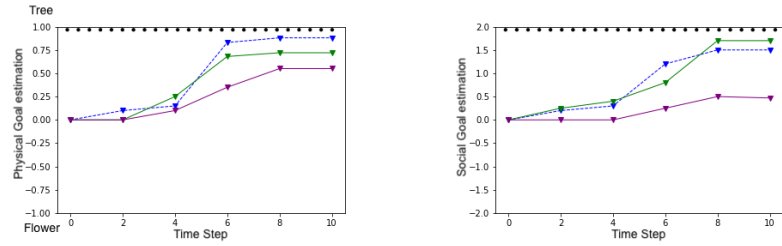
S61 Red robot is initialized with the physical goal of Tree and social goal of 0.5. Yellow robot is initialized with the physical goal of Tree and social goal of -1.



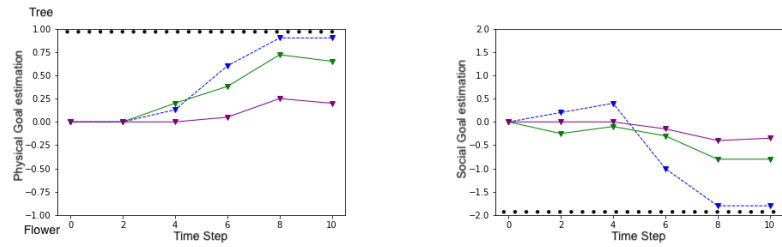
S62 Red robot is initialized with the physical goal of Tree and social goal of 1. Yellow robot is initialized with the physical goal of Tree and social goal of -1.



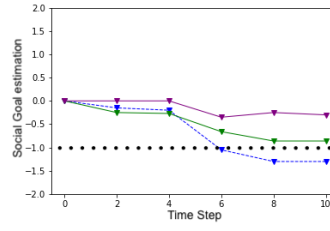
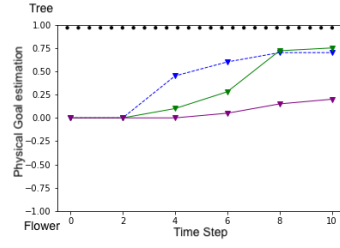
S63 Red robot is initialized with the physical goal of Tree and social goal of 2. Yellow robot is initialized with the physical goal of Tree and social goal of -1.



S64 Red robot is initialized with the physical goal of Tree and social goal of -2. Yellow robot is initialized with the physical goal of Tree and social goal of -0.5.

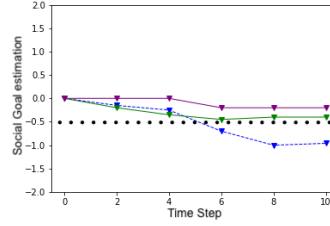
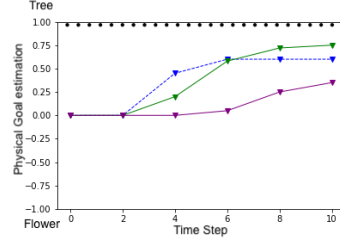


S65 Red robot is initialized with the physical goal of Tree and social goal of -1. Yellow robot is initialized with the physical goal of Tree and social goal of -0.5.



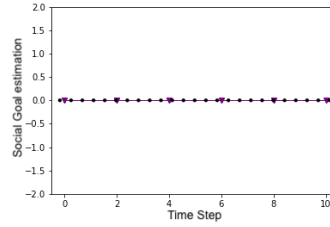
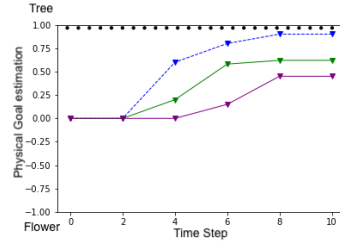
S66

Red robot is initialized with the physical goal of Tree and social goal of -0.5. Yellow robot is initialized with the physical goal of Tree and social goal of -0.5.



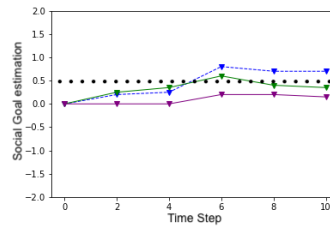
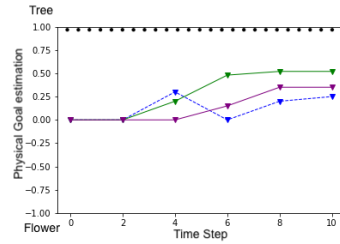
S67

Red robot is initialized with the physical goal of Tree and social goal of 0. Yellow robot is initialized with the physical goal of Tree and social goal of -0.5.



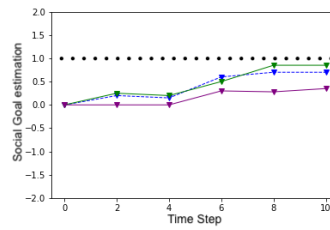
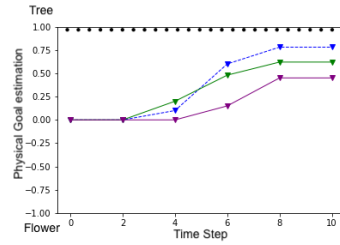
S68

Red robot is initialized with the physical goal of Tree and social goal of 0.5. Yellow robot is initialized with the physical goal of Tree and social goal of -0.5.



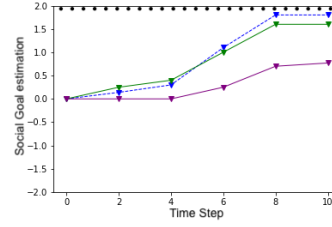
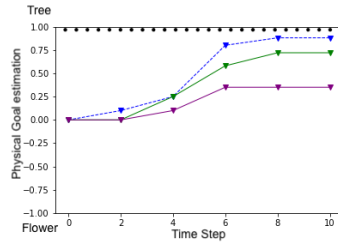
S69

Red robot is initialized with the physical goal of Tree and social goal of 1. Yellow robot is initialized with the physical goal of Tree and social goal of -0.5.



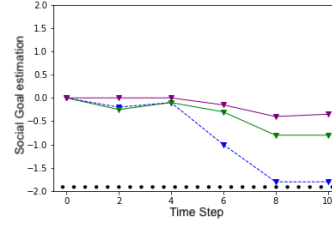
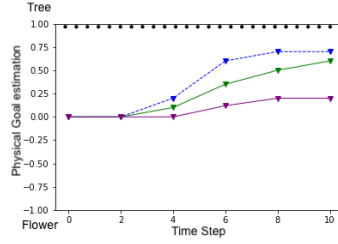
S70

Red robot is initialized with the physical goal of Tree and social goal of 2. Yellow robot is initialized with the physical goal of Tree and social goal of -0.5.



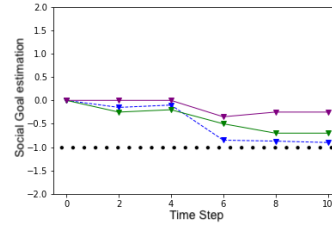
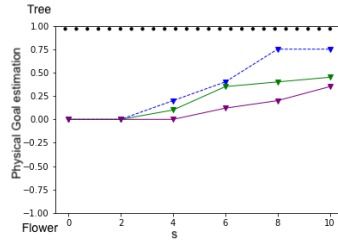
S71

Red robot is initialized with the physical goal of Tree and social goal of -2. Yellow robot is initialized with the physical goal of Tree and social goal of 0.



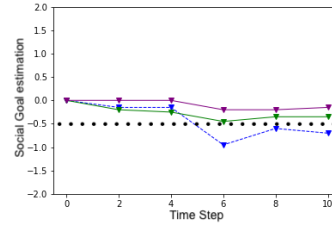
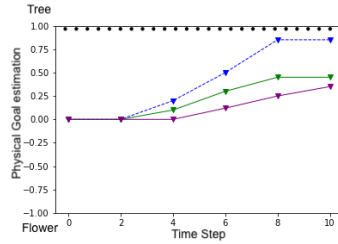
S72

Red robot is initialized with the physical goal of Tree and social goal of -1. Yellow robot is initialized with the physical goal of Tree and social goal of 0.



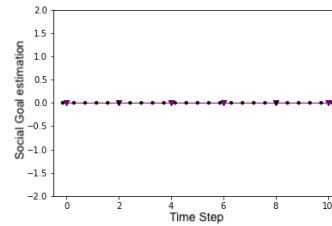
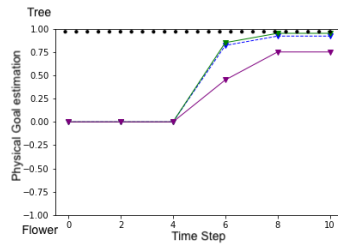
S73

Red robot is initialized with the physical goal of Tree and social goal of -0.5. Yellow robot is initialized with the physical goal of Tree and social goal of 0.



S74

Red robot is initialized with the physical goal of Tree and social goal of 0. Yellow robot is initialized with the physical goal of Tree and social goal of 0.



S75

Red robot is initialized with the physical goal of Tree and social goal of 0.5. Yellow robot is initialized with the physical goal of Tree and social goal of 0.



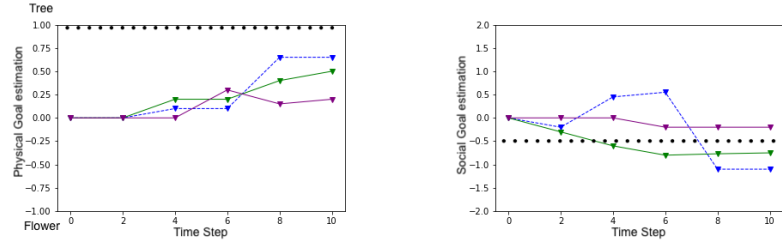
S77

S78

S79

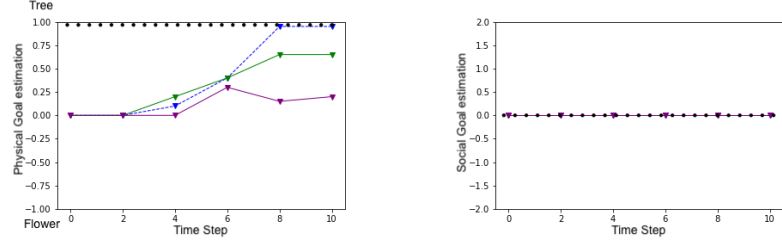
S80

28



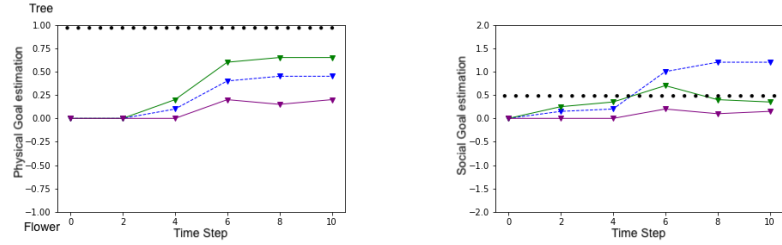
S81

Red robot is initialized with the physical goal of Tree and social goal of 0. Yellow robot is initialized with the physical goal of Tree and social goal of 0.5.



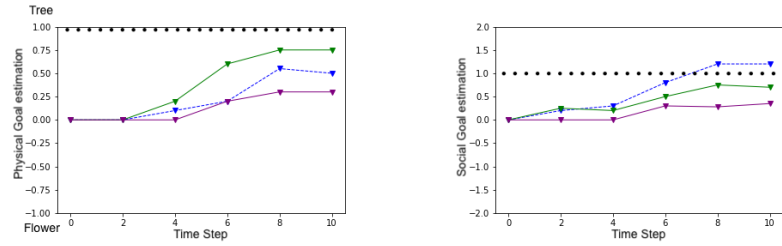
S82

Red robot is initialized with the physical goal of Tree and social goal of 0.5. Yellow robot is initialized with the physical goal of Tree and social goal of 0.5.



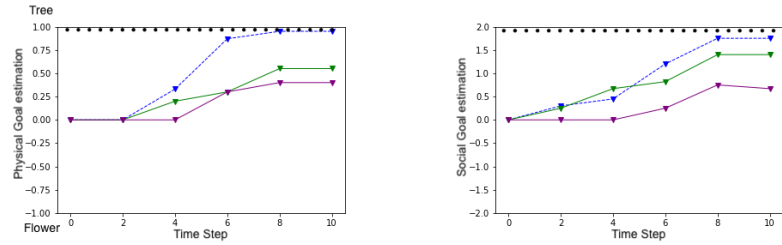
S83

Red robot is initialized with the physical goal of Tree and social goal of 1. Yellow robot is initialized with the physical goal of Tree and social goal of 0.5.



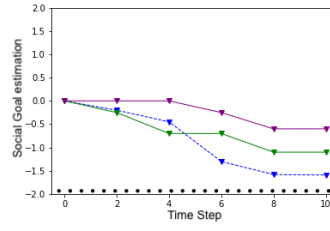
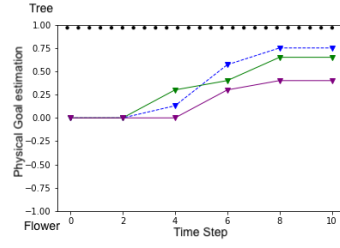
S84

Red robot is initialized with the physical goal of Tree and social goal of 2. Yellow robot is initialized with the physical goal of Tree and social goal of 0.5.



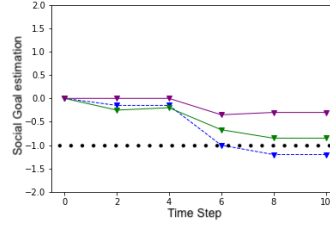
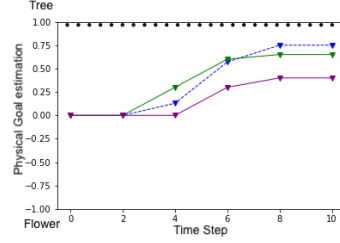
S85

Red robot is initialized with the physical goal of Tree and social goal of -2. Yellow robot is initialized with the physical goal of Tree and social goal of 1.



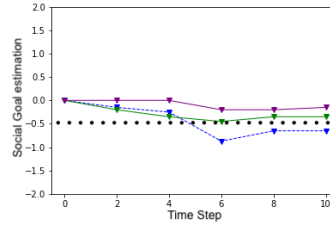
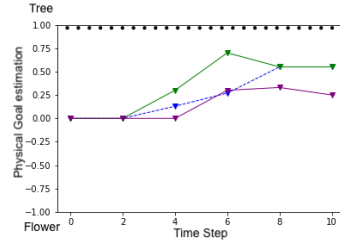
S86

Red robot is initialized with the physical goal of Tree and social goal of -1. Yellow robot is initialized with the physical goal of Tree and social goal of 1.



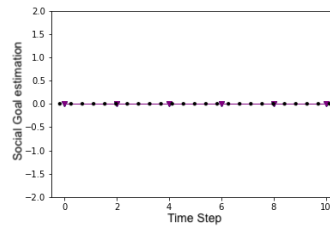
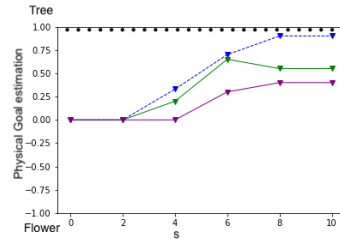
S87

Red robot is initialized with the physical goal of Tree and social goal of -0.5. Yellow robot is initialized with the physical goal of Tree and social goal of 1.



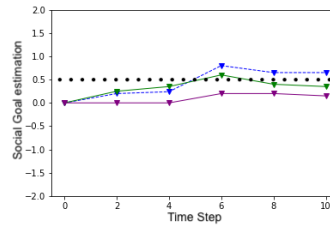
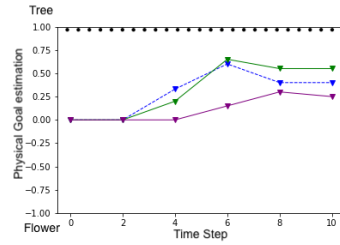
S88

Red robot is initialized with the physical goal of Tree and social goal of 0. Yellow robot is initialized with the physical goal of Tree and social goal of 1.



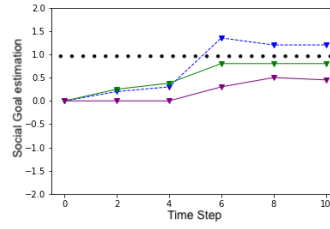
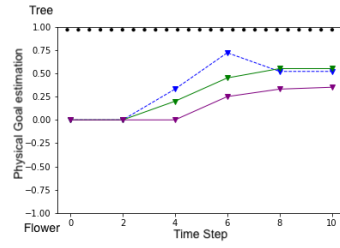
S89

Red robot is initialized with the physical goal of Tree and social goal of 0.5. Yellow robot is initialized with the physical goal of Tree and social goal of 1.



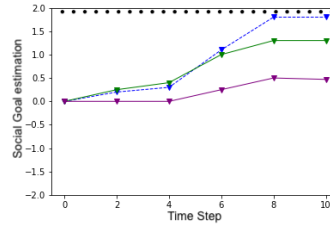
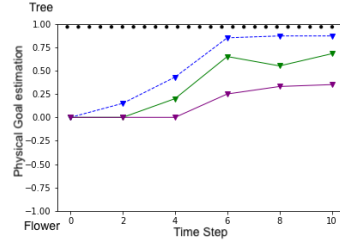
S90

Red robot is initialized with the physical goal of Tree and social goal of 1. Yellow robot is initialized with the physical goal of Tree and social goal of 1.



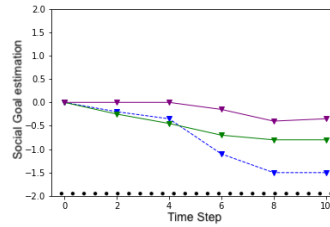
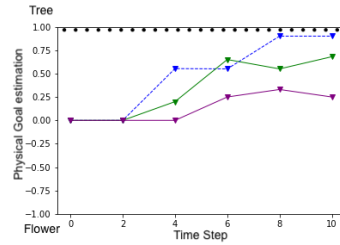
S91

Red robot is initialized with the physical goal of Tree and social goal of 2. Yellow robot is initialized with the physical goal of Tree and social goal of 1.



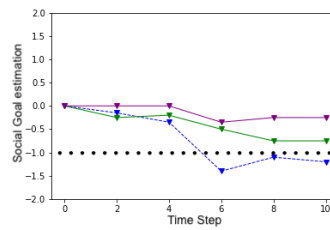
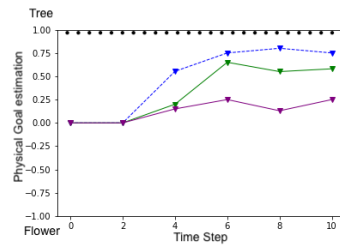
S92

Red robot is initialized with the physical goal of Tree and social goal of -2. Yellow robot is initialized with the physical goal of Tree and social goal of 2.



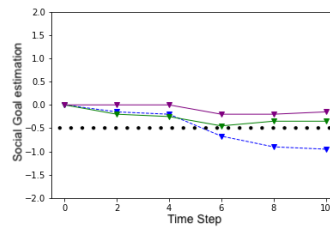
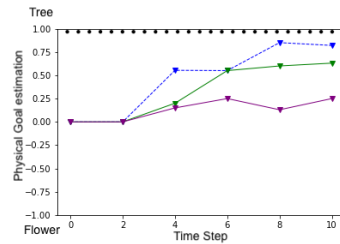
S93

Red robot is initialized with the physical goal of Tree and social goal of -1. Yellow robot is initialized with the physical goal of Tree and social goal of 2.



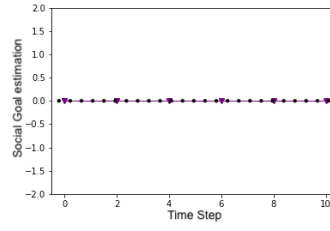
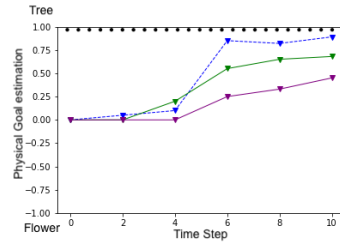
S94

Red robot is initialized with the physical goal of Tree and social goal of -0.5. Yellow robot is initialized with the physical goal of Tree and social goal of 2.



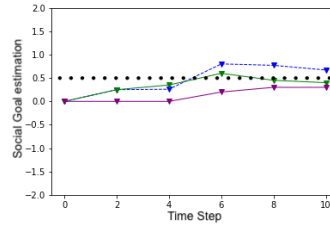
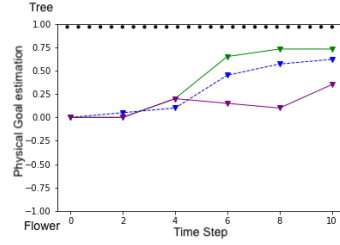
S95

Red robot is initialized with the physical goal of Tree and social goal of 0. Yellow robot is initialized with the physical goal of Tree and social goal of 2.



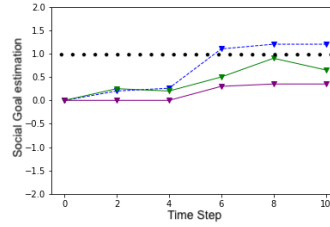
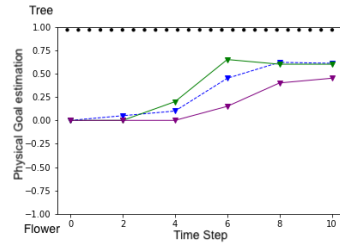
S96

Red robot is initialized with the physical goal of Tree and social goal of 0.5. Yellow robot is initialized with the physical goal of Tree and social goal of 2.



S97

Red robot is initialized with the physical goal of Tree and social goal of 1. Yellow robot is initialized with the physical goal of Tree and social goal of 2.



S98

Red robot is initialized with the physical goal of Tree and social goal of 2. Yellow robot is initialized with the physical goal of Tree and social goal of 2.

