Incorporating Rich Social Interactions Into MDPs

Ravi Tejwani*, Yen-Ling Kuo*, Tianmin Shu, Bennett Stankovits, Dan Gutfreund, Joshua B. Tenenbaum, Boris Katz, Andrei Barbu

Abstract

Much of what we do as humans is engage socially with other agents, a skill that robots must also eventually possess. We demonstrate that a rich theory of social interactions originating from microsociology and economics can be formalized by extending a nested MDP where agents reason about arbitrary functions of each other's hidden rewards. This extended Social MDP allows us to encode the five basic interactions that underlie microsociology: cooperation, conflict, coercion, competition, and exchange. The result is a robotic agent capable of executing social interactions zero-shot in new environments; like humans it can engage socially in novel ways even without a single example of that social interaction. Moreover, the judgments of these Social MDPs align closely with those of humans when considering which social interaction is taking place in an environment. This method both sheds light on the nature of social interactions, by providing concrete mathematical definitions, and brings rich social interactions into a mathematical framework that has proven to be natural for robotics, MDPs.

This material is based upon work supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216.
Incorporating Rich Social Interactions Into MDPs

Ravi Tejwani\textsuperscript{1*}, Yen-Ling Kuo\textsuperscript{1*}, Tianmin Shu\textsuperscript{2}, Bennett Stankovits\textsuperscript{1}
Dan Gutfreund\textsuperscript{3}, Joshua B. Tenenbaum\textsuperscript{2}, Boris Katz\textsuperscript{1}, Andrei Barbu\textsuperscript{1}

Abstract—Much of what we do as humans is engage socially with other agents, a skill that robots must also eventually possess. We demonstrate that a rich theory of social interactions originating from microsociology can be formalized by extending a nested MDP where agents reason about arbitrary functions of each other’s rewards. This extended Social MDP allows us to encode the five basic interactions that underlie microsociology: cooperation, conflict, coercion, competition, and exchange. The result is a robotic agent capable of executing social interactions in new environments with no interaction-specific training: like humans it can engage socially in novel ways even without a single example of that social interaction. Moreover, the estimations of these Social MDPs align closely with the judgements of humans when considering which social interaction is taking place in an environment. This method both sheds light on the nature of social interactions, by providing concrete mathematical definitions, and brings rich social interactions into a mathematical framework that has proven to be natural for robotics.

I. INTRODUCTION

Endowing robots with the ability to understand and engage in social interactions is key to having them integrate into our daily lives. Yet, very little is known about social interactions, what they are, how to measure them, and how to computationally implement them. Research in psychology has attempted to define different types of social interactions and to test the abilities of humans to engage in them \cite{1,2,3,4}. While no single framework for what social interactions are has emerged out of this work, one coherent proposal comes out of the field of microsociology; the study of fine-grained face-to-face interactions. Microsociology puts forward that underlying social interactions are five fundamental abilities that are recombined to give rise to the repertoire of behaviors we see in humans: cooperation, conflict, competition, coercion, and exchange. We provide the first computational mechanism for implementing these five interactions. Moreover, we put forward the first clear mathematical definition of what these five types of interactions are. The result is a principled computational mechanism for social interactions that allows robots to execute such interactions zero-shot, i.e. without interaction-specific training, in novel environments.

To implement these interactions, we build on top of Tejwani et al. \cite{5}, which recently introduced Social MDPs — an attempt at generalizing the MDP framework to social interactions. That work builds on an analogy, that the understanding of partial observability took a great leap forward with the advent of POMDPs, and that perhaps doing the same for social interactions by building Social MDPs will similarly pay off. In a Social MDP, instead of estimating the underlying state of the agent, it estimates the reward function of the other agent in order to help or hinder that agent. This estimation is performed recursively by solving MDPs at different levels, i.e., is the other agent helping or hindering. As executing MDPs is fairly efficient, this also ensures that the resulting models are tractable. Unfortunately, as originally formulated, Social MDPs are fundamentally limited to two of the five interactions: cooperation (helping) and conflict (hindering). Our new model is an extended Social MDP which is capable of handling these five fundamental social interactions. One feature of this model is that it degenerates to the original Social MDP model for the two interactions that the two share, and furthermore, it degenerates to a well-known model of altruism in game theory for those two interactions \cite{6}.

We make four contributions:

1) A novel Social MDP that allows robots to execute five fundamental social interactions from microsociology.

2) The first formalization of what those interactions are.

3) A computational implementation that enables robots to engage in social interactions zero-shot, without any social-interaction-specific training.

4) Extensive human experiments that validate the model and demonstrate its ability to capture human judgments about social interactions.

II. RELATED WORK

In multi-agent settings, an agent learns to reason about the goals, preferences and beliefs of the other agents so that it can effectively interact with them \cite{7,8}. Several types of models have been explored in this space: theory-of-mind-based models for goal inference \cite{9,10}, Bayesian inverse planning \cite{11,12}, and learning the reward functions of other agents \cite{13}. Xie et al. \cite{14} provide a method for learning a low-dimensional representation of the strategy of another agent. This representation enables agents to avoid or work with one another. These methods have all been limited to social interactions exhibiting cooperation (helping) or conflict (hindering). We go well beyond such models supporting a far richer theory of social interactions.

Social actions such as walking, waving, hugging, and hand-shaking in videos of group activities have also been explored \cite{15,16}. These methods broadly involve two phases \cite{17}: a social perception phase and a coordination phase where agents interact. In contrast, Social MDPs are
agnostic to specific social actions. We are not detecting if two agents are hugging and then inferring that they are friendly because we have seen that hugging often results in positive actions. We are specifying a reward for what cooperative actions are; given a novel action, that has never been seen before, in a novel context, Social MDPs determine which, if any, social interaction is at play.

In game theory, several approaches [6][22] explore altruistic and spiteful actions (the term of art for cooperation and conflict) by means of linear combinations of payoffs. Work of Levine [6] is mathematically equivalent to the level 1 Social MDPs defined in Tejwani et al. [5], where an agent estimates the other agent’s physical goal in order to help or hinder, while level 2 Social MDPs, i.e. the agent also estimates if the other agent is helping or hindering, seem to have no counterpart in the current economics literature. The level-2 Social MDPs is the equivalent of saying that in a game you can help another player, but you cannot help another player whose goal is to then help a third player. Our long-term goal is to connect the extended Social MDPs presented here back to the economics literature and expand both the depth of inference considered there and the range of social interactions.

Human social perception has been studied extensively, but with limited theoretical insights. A rich history of work with animations of simple geometrical objects [23][24] shows that humans readily assign social goals and intentions to any agent. More recently, agents’ behaviors have been simulated with modern physics engines in fully observable [25][26] and partially observable environments [27][21] to create various social benchmarks, although these benchmarks are almost exclusively limited to helping and hindering. We hope that by providing formal definitions for what social interactions are, we can eventually bring about a much closer connection between the cognitive science and robotics of social interactions.

### III. Model

We adopt the setting of Tejwani et al. [5] who initially develop Social MDPs by giving agents a combination of physical and social goals. A physical goal is precisely what an MDP can represent, a reward that is a function of the state of the world. A social goal extends the reward function to allow it to contain an estimate of the reward function of another agent. In the original Social MDP formulation, every agent estimated the rewards of every other agent. Then, the reward of each agent was a linear combination of its own physical goal and its estimate of the reward function of other agents (the social goal).

The coefficient of this linear combination of physical and social goals determined how willing the agent was to engage socially. If it was zero, no weight was put on the social goal, so the agent behaved as an MDP. A large positive value gives an agent a high reward when another agent maximizes its own reward — the result is an agent that helps the other agent succeed. A large negative value does the opposite, gives an agent a low reward when another agent maximizes its own reward — the result is an agent that stops the other agent from succeeding. Tejwani et al. [5] demonstrated that this formulation gives rise to robots that appear to humans to behave socially.

That framework lacks any degrees of freedom that would allow it to represent social interactions other than helping and hindering. It has a single coefficient that determines the interaction between agents and the polarity of that coefficient determines if the interaction is helpful or unhelpful.

Next, we survey a precise mathematical definition of the five social interaction types considered in this framework of recursively estimating other agent’s social reward functions and then demonstrate how to extend Social MDPs to enable robots to execute these interactions.

#### A. The five interaction types

To define the pairwise interactions between agent $i$ and $j$, let the reward of an agent $i$ at level $l$, $R^i_l$, be a combination of that agent’s physical goal, $g_i$ and that agent’s social goal, $ξ_{ij}^l$: $R^i_l = r(g_i) + ξ_{ij}^l$. Let $ξ_{ij}^l$ be the social goal of agent $i$ toward agent $j$ at level $l$. At $l = 0$, no agents are social, the Social MDP degenerates into a plain MDP, i.e., $ξ_{ij}^0 = 0$. At $l = 1$, each agent has a social goal, but it does not consider other agents to have social goals, only physical goals. At $l = 2$, agents assume that other agents also have social goals, etc. This nested formulation allows Social MDPs to start with an arbitrary level, then eventually bottom out in an MDP. Next, we will show how the reward function of agent $i$ is formulated when that agent engages in each of the five social interaction types.

We define all interactions from the perspective of agent $i$, i.e., if agent $i$ wants to be social with respect to agent $j$ in each of these five ways, what is $i$’s reward function? Reward functions are specified in terms of estimated rewards, as $i$ does not know what $j$ wants, it must infer $j$’s reward function; we denote all estimated quantities with a tilde, as in $R^i_{l-1}$ for the estimate of $j$’s reward function made by $i$ at level $l - 1$ and $ξ^i_{ij}l$ for the estimate of the social interaction between $i$ and $j$ at level $l$ again made by $i$. In each case, the additional superscript $i$ denotes the agent performing the estimation — agent $i$ and agent $k$ may, depending on their observations or biases, not estimate the same reward function for agent $j$.

An overview of the recursive inferences being made between two agents is shown in Fig. [1] In Table [1] we show the social component of the reward function for agent $i$ as a function of its own physical goal and the estimated social goal of $j$ when $l > 1$ (when other agents are assumed to have social goals) and when $l = 0$ (when other agents have only physical goals). For each agent, we estimate what kind of social interaction it is engaging in. Then, the reward function of the agent is updated according to Fig. [1] For example, if a level 2 agent $i$ intends to cooperate with $j$, the original reward function template for $i$, $R^i_l = r(g_i) + ξ_{ij}^l$, is updated as $R^i_l = r(g_i) + R^j_{l-1}$ — agent $i$’s reward now includes agent $j$’s reward so $i$ will help $j$. The coefficient of this linear combination of physical and social goals determined how willing the agent was to engage socially. If it was zero, no weight was put on the social goal, so the agent behaved as an MDP. A large positive value gives an agent a high reward when another agent maximizes its own reward — the result is an agent that helps the other agent succeed. A large negative value does the opposite, gives an agent a low reward when another agent maximizes its own reward — the result is an agent that stops the other agent from succeeding. Tejwani et al. [5] demonstrated that this formulation gives rise to robots that appear to humans to behave socially.
Fig. 1: Reasoning with an extended Social MDP rendered as Bayesian network at different levels. $s$ is the observed state, $\psi$ is the policy, $g$ is the physical goal, and $\xi$ is a social goal. The yellow robot, agent 1, is reasoning at level two about the red robot’s (agent 2) goals – not only what physical goal the red agent is reaching, i.e. a tree, but also what social goal the red agent has, i.e. cooperating with the yellow. This demonstrates the utility of nesting Social MDPs, shallower social MDPs are more limited in their reasoning about the social abilities of other agents.

Algorithm 1 The algorithm to compute the social policy $\psi_{ij}^l$ for agent $i$ at level $l$ and time $t$. We use the estimated social policy $\tilde{\psi}_{ij}^l$ at the previous time step to update the estimated rewards as described in Section III-C. At $t = 0$, we assume $P(g_{ij}^{-1,0})$ and $P(\xi_{ij}^{-1,0})$ are sampled from uniform distributions. This algorithm is called at all recursion steps $\tilde{\psi}_{ij}^{l-1}$ to estimate social policy for the other agent $j$. The estimated reward and policy are used to compute the $Q$ values for action selection.

**Require:** $l$, $s_i^t$, $a_i^t$, $a_j^t$, $\xi_{ij}$, $g_i$

if $l = 0$ then

solve MDP for agent $i$

else

for all $\tilde{\xi}_{ij}^{-1,t}$, $\tilde{g}_{ij}^{-1,t}$ do

compute $P(\xi_{ij}^{-1,t} | s_i^{-1}, a_i^{-1}, a_j^{-1})$ and $P(g_{ij}^{-1,t} | s_i^{-1}, a_i^{-1}, a_j^{-1})$

$\tilde{\psi}_{ij}^{l-1} (s_i^t, a_i^t, a_j^t, \tilde{\xi}_{ij}^{-1,t}, \tilde{g}_{ij}^{-1,t})$

end for

compute $R_i(s_i^t, a_i^t, a_j^t, \xi_{ij}, g_i)$

compute $Q_i(s_i^t, a_i^t, a_j^t, \xi_{ij}, g_i)$

$\pi_i^t \leftarrow \arg \max_{a_i \in A_i} Q_i$

end if

TABLE I: Computing the reward function of a social agent $i$, engaging with agent $j$ in one of five social interactions at level $l$. Definitions for what these concepts refer to in our framework as provided; this is far from covering every nuance of these social interactions. The reward function for $i$ is of the form $R_i^t = r(g_i^t) + \xi_{ij}^t$, where $r(g_i^t)$ is the reward for the physical goal of agent $i$ (the reward in a plain MDP) and $\xi_{ij}^t$ is the social interaction of $i$ toward $j$. Based on the type of interaction that $i$ wants to have with $j$ and its estimate of what the reciprocal interaction is ($i$’s estimate of how $j$ wants to interact with $i$, $\tilde{\xi}_{ij}^t$) we show what term to substitute $\xi_{ij}^t$ with achieve that interaction. For example, in cooperation, we substitute $i$’s estimate of what $j$ wants into $i$’s reward function; this gives $i$ a reward whenever $j$ accomplishes their goals. N/A are impossible interactions. See Section III-A for explanations of the other substitutions.

Cooperation and Conflict, the two types of goals supported by the original Social MDPs are peculiar: the reward function does not depend on what kind of social interaction the other agent wants to engage in. This is why they could be implemented in the original Social MDP formulation. The other three types of social interaction are more complex, they depend on the other agent’s intentions. For example, competing with someone who intends to cooperate with you is very different from competing with someone who intends to stop you from taking actions.

See Table I for the complete mapping from interactions to rewards. Competition and Coercion follow a pattern that
is similar to cooperation and conflict with several special cases since they are situation-specific. In other words, what it means to cooperate depends on what the other agent wants to do: if it is cooperative, you incorporate its reward differently into yours than if it is combative.

*Exchange* is the outlier. Exchanging implies that an agent performs an action that is useful for another and vice versa. If one of the two agents isn’t willing to exchange, it’s impossible for either to do so on their own; hence the N/A in Table I.

When directing an agent to exchange with another, we update where is the sum of immediate reward and the expected value in the future.

\[
Q_i(s, a, j, g_j) = R(s, a, j, g_j) + \gamma \sum_{s' \in S} P(s') Q_i(s', a_j, g_j)
\]

Since agent \( i \) is interacting with agent \( j \), it needs to estimate what actions the other agent is likely to take in order to compute its state-action value. Social MDPs take the expectation over the estimated goals and actions of agent \( j \) to compute \( V_j^l(s', \xi_j, g_j) \):

\[
V_j^l(s', \xi_j, g_j) = \max_{a_j \in A_j} \left\{ E_{\xi_j} [Q_i(s', a_j, \xi_j, g_j)] \right\}
\]

\[
= \max_{a_j \in A_j} \left\{ \sum_{s_j \in S_j} P(s_j | s') \sum_{a_j} P(a_j | s_j) \sum_{g_j} \sum_{\xi_j} P(\xi_j | s_j) \sum_{g_j} P(g_j | s_j) V_j^l(s_j, a_j, \xi_j, g_j) \right\}
\]

When solving agent \( i \)’s MDP at level \( l \), the estimated social and physical goals are further used to update the other agent \( j \)’s policy which generates the actions agent \( j \) may take given a state. We refer to this policy as social policy because this policy makes it possible for the agent to interact socially. We denote the estimated social policy for agent \( j \) at reasoning level \( l - 1 \) as \( \tilde{\psi}_j^{l-1} : S \times A_i \times A_j \times \tilde{\xi}_j^{l-1} \times \tilde{g}_j^{l-1} \rightarrow [0,1] \).

Algorithm I summarizes the steps to compute the state-action values and select optimal actions for any level \( l \) at time step \( t \). We first update the probability of the estimated goals of the other agent using the observed state and the estimated policy from the previous time step. The updated probability of goals are used to update the policy of the other agent and compute the reward and Q function of the target agent.

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

\[
P(\tilde{\xi}_j^{l-1} | s^{l-1}, a_j^{l-1}) \propto P(\tilde{\xi}_j^{l-1} | s^{l-2}, a_j^{l-2}, a_j^{l-3})
\]

The physical goal \( g_j \) of agent \( j \) is estimated by \( i \), similar to [25] but marginalized over the estimated social goal as the agent is estimating the social goal at the same time.

\[
P(g_j^{l-1} | s^{l-1}) \propto \int_{\tilde{\xi}_j^{l-1}} P(s^{l-1} | g_j^{l-1}) P(\tilde{\xi}_j^{l-1} | g_j^{l-1})
\]

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

\[
\tilde{\psi}_j^{l-1}(s, a_j, g_j) = \text{Softmax}(Q_j^{l-1}(s, a_j, g_j))
\]
We apply our extended Social MDP framework to a multi-agent grid world inspired by previous studies on social perception [15, 19, 28, 27]. This 10 × 10 world consists of two agents (a yellow robot and red robot), two physical landmarks (a construction site and a tree) and three objects (an axe, wooden log, and a water bucket). Objects can be pushed to either destination. Physical goals consist of moving the desired objects to one of the landmarks. Agents can have no social goal or one of the five social goals: cooperation, conflict, competition, coercion, or exchange. At any point in time, agents can push an object forward, move in one of the four cardinal directions or choose to take no actions.

Each agent’s reward for reaching its physical goal is based on that agent’s geodesic distance from the goal after taking an action [15]. This physical reward function is parameterized by $\rho$ and $\delta$ that determines the scale and shape of the physical reward: 
$$r(s, a, g_i) = \max(\rho (1 - \text{distance}(s, a, g_i)/\delta), 0)$$
We set the cost, $c(a)$, of an action $a$, to 1 for grid moves and to 0.1 for staying in place while $\rho$ and $\delta$ were set to 1.25 and 5, respectively. The discount factor, $\gamma$, was set to 0.99.

We systematically enumerate the 72 unique scenarios by assigning agents either the same or different physical goal, and one of five social goals or no social goal ($2 \times 6 \times 6 = 72$ scenarios). Even this simple world gives rise to many rich social interactions with subtle but meaningful differences in the behaviors of agents depending on their understanding of social interactions. Take the scenario where both agents have the social goal of exchanging something as an example. An exchange is practical only in some cases: the agents must be arranged in such a way that it would be helpful for the red agent to aid the yellow agent and vice versa (for example, the red agent is closer to the axe which the yellow agent wants and the yellow agent is closer to the log which the red agent wants). The agents must recognize this and recognize if the other agent is willing to exchange by attempting the exchange. Then, they must follow through with the exchange, bring the other object to a meeting point, swap objects, and then go on to complete their own physical goals. If one of the agents is uncooperative, both agents should abort and try to perform their own physical goals, as the exchange is then impossible. This occurs without pre-specifying any symbolic notion of “exchange”, without a single training example of an exchange, and without any hard-coded rules, merely by specifying the correct reward function.

All scenarios with descriptions, diagrams, videos, and detailed timestep-by-timestep results for all human experiments and models are available on this website: https://social-interactions-mdp.github.io

### A. Are these social interactions?

We first establish that these scenarios do indeed show social interactions and that humans are able to recognize the intended social interactions. Each of the 72 scenarios were executed with level 2 Social MDPs guiding the yellow agent giving rise to 72 videos. 12 subjects were recruited on Mechanical Turk. Subjects were instructed to rate each of the 72 videos (randomized order) according to how confident they are (between 0 to 100) that a video depicts a given social interaction, according to their own interpretation of what each of the social interactions mean (they were only given the 5 labels without any descriptions). Taking the subject’s last judgment, at which point they have seen the entire interaction, we ask how well does the most confident label that humans give to the interaction match the intended interaction from the Social MDP. Humans provided the top label with 82% accuracy, showing that Social MDPs generate the intended social interactions the vast majority of the time, see Table II

### B. Do we model human judgments?

Going further, we demonstrate that not only do humans recognize the intended social interactions from Social MDPs, but Social MDPs also predict human judgments. Taking each of the 72 videos, we compare the confidence of humans that

<table>
<thead>
<tr>
<th>Model</th>
<th>Cooperation</th>
<th>Conflict</th>
<th>Competition</th>
<th>Coercion</th>
<th>Exchange</th>
<th>Overall</th>
<th>Physical Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>0.934</td>
<td>0.952</td>
<td>0.623</td>
<td>0.724</td>
<td>0.876</td>
<td>0.823</td>
<td>0.974</td>
</tr>
<tr>
<td>Extended Social MDP (Ours)</td>
<td>0.845</td>
<td>0.851</td>
<td>0.471</td>
<td>0.651</td>
<td>0.814</td>
<td>0.726</td>
<td>0.831</td>
</tr>
<tr>
<td>Inverse Planning</td>
<td>0.763</td>
<td>0.784</td>
<td>0.261</td>
<td>0.283</td>
<td>0.197</td>
<td>0.457</td>
<td>0.783</td>
</tr>
<tr>
<td>Cue Based Model</td>
<td>0.461</td>
<td>0.434</td>
<td>0.127</td>
<td>0.156</td>
<td>0.083</td>
<td>0.252</td>
<td>0.432</td>
</tr>
</tbody>
</table>

### IV. RESULTS

Going further, we demonstrate that not only do humans recognize the intended social interactions from Social MDPs, but Social MDPs also predict human judgments. Taking each of the 72 videos, we compare the confidence of humans that...
C. Models

We compare Social MDPs with two alternative models, inverse planning [15] and a recent cue-based model [25]. Each model is provided with every video, frame by frame, and is required to incrementally predict which social interaction is taking place and what the physical goal is. We use the probability estimation from eq. 5 and 6 as the prediction from our extended Social MDPs. At our website, we list every scenario with detailed results from each model at every time step. The results for several example scenarios are shown in Fig. 3. Qualitatively, one can observe that the Social MDPs track human judgments about these videos far more accurately. Quantitatively, results aggregated across the different categories are shown in Table II. While humans are able to estimate social goals and physical goals more accurately than any model, the gap between Social MDPs and other models is stark. Not only do Social MDPs estimate the physical goal more accurately, but the social goal accuracy is dramatically higher (72.6% vs 45.7%). Our online supplement also demonstrates the different levels of nesting.

V. Conclusion

We demonstrated extended Social MDPs which are able to reason about the five core social interactions described in the microsociology literature. This model is capable of reasoning about social interactions without requiring interaction-specific training, just like humans are able to recognize social interactions in novel environments. The inferences the model makes both qualitatively and quantitatively match human judgments. This significantly expands the space of social interactions that robots engage in. The runtime of the model depends on the number of levels considered as this controls how many MDPs must be solved recursively. We do not yet know which level corresponds to the inferences that humans are able to make.

We would like to scale this approach up to continuous environments and to physical robots with manipulators. In addition, we would like to recognize these social interactions from real-world videos rather than from renderings and trajectories of agents. Hopefully, sometime in the future, robots will have general-purpose social skills and the field of studying social interactions will be based on mathematical and refutable theories that span robotics, social science, cognitive science, vision, and neuroscience.

ACKNOWLEDGMENTS

This work was supported by the Center for Brains, Minds and Machines, NSF STC award 1231216, the MIT CSAIL Systems that Learn Initiative, the CBMM-Siemens Graduate Fellowship, the MIT-IBM Watson AI Lab, the DARPA Artificial Social Intelligence for Successful Teams (ASIST) program, the United States Air Force Research Laboratory and United States Air Force Artificial Intelligence Accelerator under Cooperative Agreement Number FA8750-19-2-1000, and the Office of Naval Research under Award Number N00014-20-1-2589 and Award Number N00014-20-1-2643. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.