Social Influence as Intrinsic Motivation for Multi-Agent Deep Reinforcement Learning

Natasha Jaques [1 2] **Angeliki Lazaridou** [2] **Edward Hughes** [2] **Caglar Gulcehre** [2] **Pedro A. Ortega** [2] **DJ Strouse** [3] **Joel Z. Leibo** [2] **Nando de Freitas** [2]

Abstract

We propose a unified mechanism for achieving coordination and communication in Multi-Agent Reinforcement Learning (MARL), through rewarding agents for having causal influence over other agents' actions. Causal influence is assessed using counterfactual reasoning. At each timestep, an agent simulates alternate actions that it could have taken, and computes their effect on the behavior of other agents. Actions that lead to bigger changes in other agents' behavior are considered influential and are rewarded. We show that this is equivalent to rewarding agents for having high mutual information between their actions. Empirical results demonstrate that influence leads to enhanced coordination and communication in challenging social dilemma environments, dramatically increasing the learning curves of the deep RL agents, and leading to more meaningful learned communication protocols. The influence rewards for all agents can be computed in a decentralized way by enabling agents to learn a model of other agents using deep neural networks. In contrast, key previous works on emergent communication in the MARL setting were unable to learn diverse policies in a decentralized manner and had to resort to centralized training. Consequently, the influence reward opens up a window of new opportunities for research in this area.

1. Introduction

Intrinsic Motivation for Reinforcement Learning (RL) refers to reward functions that allow agents to learn useful behavior across a variety of tasks and environments, sometimes in the absence of environmental reward (Singh et al., 2004). Previous approaches to intrinsic motivation often focus on curiosity (e.g. Pathak et al. (2017); Schmidhuber (2010)), or empowerment (e.g. Klyubin et al. (2005); Mohamed & Rezende (2015)). Here, we consider the problem of deriving intrinsic social motivation from other agents in multi-agent RL (MARL). Social learning is incredibly important for humans, and has been linked to our ability to achieve unprecedented progress and coordination on a massive scale (Henrich, 2015; Harari, 2014; Laland, 2017; van Schaik & Burkart, 2011; Herrmann et al., 2007). While some previous work has investigated intrinsic social motivation for RL (e.g. Sequeira et al. (2011); Hughes et al. (2018); Peysakhovich & Lerer (2018)), these approaches rely on hand-crafted rewards specific to the environment, or allowing agents to view the rewards obtained by other agents. Such assumptions make it impossible to achieve independent training of MARL agents across multiple environments.

Achieving coordination among agents in MARL still remains a difficult problem. Prior work in this domain (e.g., Foerster et al. (2017; 2016)), often resorts to centralized training to ensure that agents learn to coordinate. While communication among agents could help with coordination, training emergent communication protocols also remains a challenging problem; recent empirical results underscore the difficulty of learning meaningful emergent communication protocols, even when relying on centralized training (e.g., Lazaridou et al. (2018); Cao et al. (2018); Foerster et al. (2016)).

We propose a unified method for achieving both coordination and communication in MARL by giving agents an intrinsic reward for having a causal influence on other agents' actions. Causal influence is assessed using counterfactual reasoning; at each timestep, an agent simulates alternate, counterfactual actions that it could have taken, and assesses their effect on another agent's behavior. Actions that lead to relatively higher change in the other agent's behavior are considered to be highly influential and are rewarded. We show how this reward is related to maximizing the mutual information between agents' actions, and hypothesize that this inductive bias will drive agents to learn coordinated behavior. Maximiz-

[1]Media Lab, Massachusetts Institute of Technology, Cambridge, USA [2]Google DeepMind, London, UK [3]Princeton University, Princeton, USA. Correspondence to: Natasha Jaques <jaquesn@mit.edu>, Angeliki Lazaridou <angeliki@google.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

ing mutual information as a form of intrinsic motivation has been studied in the literature on empowerment (e.g. Klyubin et al. (2005); Mohamed & Rezende (2015)). Social influence can be seen as a novel, social form of empowerment.

To study our influence reward, we adopt the Sequential Social Dilemma (SSD) multi-agent environments of Leibo et al. (2017). Through a series of three experiments, we show that the proposed social influence reward allows agents to learn to coordinate and communicate more effectively in these SSDs. We train recurrent neural network policies directly from pixels, and show in the first experiment that deep RL agents trained with the proposed social influence reward learn effectively and attain higher collective reward than powerful baseline deep RL agents, which often completely fail to learn.

In the second experiment, the influence reward is used to directly train agents to use an explicit communication channel. We demonstrate that the communication protocols trained with the influence reward are more meaningful and effective for obtaining better collective outcomes. Further, we find a significant correlation between being influenced through communication messages and obtaining higher individual reward, suggesting that influential communication is beneficial to the agents that receive it. By examining the learning curves in this second experiment, we again find that the influence reward is essential to allow agents to learn to coordinate.

Finally, we show that influence agents can be trained independently, when each agent is equipped with an internal neural network *Model of Other Agents* (MOA), which has been trained to predict the actions of every other agent. The agent can then simulate counterfactual actions and use its own internal MOA to predict how these will affect other agents, thereby computing its own intrinsic influence reward. Influence agents can thus learn socially, only through observing other agents' actions, and without requiring a centralized controller or access to another agent's reward function. Therefore, the influence reward offers us a simple, general and effective way of overcoming long-standing unrealistic assumptions and limitations in this field of research, including centralized training and the sharing of reward functions or policy parameters. Moreover, both the influence rewards as well as the agents' policies can be learned directly from pixels using expressive deep recurrent neural networks. In this third experiment, the learning curves once again show that the influence reward is essential for learning to coordinate in these complex domains.

The paper is structured as follows. We describe the environments in Section 2, and the MARL setting in Section 3. Section 4 introduces the basic formulation of the influence reward, Section 5 extends it with the inclusion of explicit communication protocols, and Section 6 advances it by including models of other agents to achieve independent training. Each of these three sections presents experiments and results that empirically demonstrate the efficacy of the social influence reward. Related work is presented in Section 7. Finally, more details about the causal inference procedure are given in Section 8.

2. Sequential Social Dilemmas

Sequential Social Dilemmas (SSDs) (Leibo et al., 2017) are partially observable, spatially and temporally extended multi-agent games with a game-theoretic payoff structure. An individual agent can obtain higher reward in the short-term by engaging in defecting, non-cooperative behavior (and thus is greedily motivated to defect), but the total payoff per agent will be higher if all agents cooperate. Thus, the collective reward obtained by a group of agents in these SSDs gives a clear signal about how well the agents learned to cooperate (Hughes et al., 2018).

We experiment with two SSDs in this work, a public goods game *Cleanup*, and a public pool resource game *Harvest*. In both games apples (green tiles) provide the rewards, but are a limited resource. Agents must coordinate harvesting apples with the behavior of other agents in order to achieve cooperation (for further details see Section 2 of the Supplementary Material). For reproducibility, the code for these games has been made available in open-source.[1]

As the Schelling diagrams in Figure 10 of the Supplementary Material reveal, all agents would benefit from learning to cooperate in these games, because even agents that are being exploited get higher reward than in the regime where more agents defect. However, traditional RL agents struggle to learn to coordinate or cooperate to solve these tasks effectively (Hughes et al., 2018). Thus, these SSDs represent challenging benchmark tasks for the social influence reward. Not only must influence agents learn to coordinate their behavior to obtain high reward, they must also learn to cooperate.

3. Multi-Agent RL for SSDs

We consider a MARL Markov game defined by the tuple $\langle S,T,A,r\rangle$, in which multiple agents are trained to independently maximize their own individual reward; agents do not share weights. The environment state is given by $s \in S$. At each timestep t, each agent k chooses an action $a_t^k \in A$. The actions of all N agents are combined to form a joint action $\boldsymbol{a}_t = [a_t^0,...a_t^N]$, which produces a transition in the environment $T(s_{t+1}|\boldsymbol{a}_t,s_t)$, according to the state transition distribution T. Each agent then receives its own reward $r^k(\boldsymbol{a}_t,s_t)$, which may depend on the actions of other agents. A history of these variables over time is termed a trajectory, $\tau = \{s_t,\boldsymbol{a}_t,\boldsymbol{r}_t\}_{t=0}^T$. We consider a partially observable

[1] https://github.com/eugenevinitsky/sequential_social_dilemma_games

setting in which the kth agent can only view a portion of the true state, s_t^k. Each agent seeks to maximize its own total expected discounted future reward, $R^k = \sum_{i=0}^{\infty} \gamma^i r_{t+i}^k$, where γ is the discount factor. A distributed asynchronous advantage actor-critic (A3C) approach (Mnih et al., 2016) is used to train each agent's policy π^k.

Our neural networks consist of a convolutional layer, fully connected layers, a Long Short Term Memory (LSTM) recurrent layer (Gers et al., 1999), and linear layers. All networks take images as input and output both the policy π^k and the value function $V^{\pi_k}(s)$, but some network variants consume additional inputs and output either communication policies or models of other agents' behavior. We will refer to the internal LSTM state of the kth agent at timestep t as u_t^k.

4. Basic Social Influence

Social influence intrinsic motivation gives an agent additional reward for having a causal influence on another agent's actions. Specifically, it modifies an agent's immediate reward so that it becomes $r_t^k = \alpha e_t^k + \beta c_t^k$, where e_t^k is the extrinsic or environmental reward, and c_t^k is the causal influence reward.

To compute the causal influence of one agent on another, suppose there are two agents, k and j, and that agent j is able to condition its policy on agent k's action at time t, a_t^k. Thus, agent j computes the probability of its next action as $p(a_t^j | a_t^k, s_t^j)$. We can then intervene on a_t^k by replacing it with a counterfactual action, $\tilde{a}_t^k$. This counterfactual action is used to compute a new distribution over j's next action, $p(a_t^j | \tilde{a}_t^k, s_t^j)$. Essentially, agent k asks a retrospective question: "How would j's action change if I had acted differently in this situation?".

By sampling several counterfactual actions, and averaging the resulting policy distribution of j in each case, we obtain the marginal policy of j, $p(a_t^j | s_t^j) = \sum_{\tilde{a}_t^k} p(a_t^j | \tilde{a}_t^k, s_t^j) p(\tilde{a}_t^k | s_t^j)$ —in other words, j's policy if it did not consider agent k. The discrepancy between the marginal policy of j and the conditional policy of j given k's action is a measure of the causal influence of k on j; it gives the degree to which j changes its planned action distribution because of k's action. Thus, the causal influence reward for agent k is:

$$c_t^k = \sum_{j=0, j\neq k}^{N} \Big[D_{KL}\big[p(a_t^j \,|\, a_t^k, s_t^j) \,\Big\|\, \sum_{\tilde{a}_t^k} p(a_t^j \,|\, \tilde{a}_t^k, s_t^j) p(\tilde{a}_t^k \,|\, s_t^j) \big] \Big]$$

$$= \sum_{j=0, j\neq k}^{N} \Big[D_{KL}\big[p(a_t^j \,|\, a_t^k, s_t^j) \,\Big\|\, p(a_t^j \,|\, s_t^j) \big] \Big]. \quad (1)$$

Note that it is possible to use a divergence metric other than KL; we have found empirically that the influence reward is robust to the choice of metric.

The reward in Eq. 4 is related to the mutual information (MI) between the actions of agents k and j, $I(a^k; a^j | s)$. As the reward is computed over many trajectories sampled independently from the environment, we obtain a Monte-Carlo estimate of $I(a^k; a^j | s)$. In expectation, the influence reward incentivizes agents to maximize the mutual information between their actions. The proof is given in Section 10.1 of the Supplementary Material. Intuitively, training agents to maximize the MI between their actions results in more coordinated behavior.

Moreover, the variance of policy gradient updates increases as the number of agents in the environment grows (Lowe et al., 2017). This issue can hinder convergence to equilibrium for large-scale MARL tasks. Social influence can reduce the variance of policy gradients by introducing explicit dependencies across the actions of each agent. This is because the conditional variance of the gradients an agent is receiving will be less than or equal to the marginalized variance.

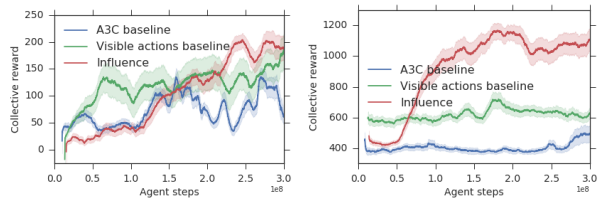
Note that for the basic influence model we make two assumptions: 1) we use centralized training to compute c_t^k directly from the policy of agent j, and 2) we assume that influence is unidirectional: agents trained with the influence reward can only influence agents that are not trained with the influence reward (the sets of influencers and influencees are disjoint, and the number of influencers is in $[1, N-1]$). Both of these assumptions are relaxed in later sections. Further details, as well as further explanation of the causal inference procedure (including causal diagrams) are available in Section 8.

4.1. Experiment I: Basic Influence

Figure 1 shows the results of testing agents trained with the basic influence reward against standard A3C agents, and an ablated version of the model in which agents do not receive the influence reward, but are able to condition their policy on the actions of other agents (even when the other agents are not within the agent's partially observed view of the environment). We term this ablated model the *visible actions baseline*. In this and all other results figures, we measure the total collective reward obtained using the best hyperparameter setting tested with 5 random seeds each. Error bars show a 99.5% confidence interval (CI) over the random seeds, computed within a sliding window of 200 agent steps. We use a curriculum learning approach which gradually increases the weight of the social influence reward over C steps ($C \in [0.2 - 3.5] \times 10^8$); this sometimes leads to a slight delay before the influence models' performance improves.

As is evident in Figures 1a and 1b, introducing an awareness of other agents' actions helps, but having the social influence reward eventually leads to significantly higher collective reward in both games. Due to the structure of the SSD games, we can infer that agents that obtain higher collective reward learned to cooperate more effectively. In the *Harvest* MARL

setting, it is clear that the influence reward is essential to achieve any reasonable learning.



(a) *Cleanup* (b) *Harvest*

Figure 1: Total collective reward obtained in Experiment 1. Agents trained with influence (red) significantly outperform the baseline and ablated agents. In Harvest, the influence reward is essential to achieve any meaningful learning.

To understand how social influence helps agents achieve cooperative behavior, we investigated the trajectories produced by high scoring models in both *Cleanup* and *Harvest*; the analysis revealed interesting behavior. As an example, in the *Cleanup* video available here: https://youtu.be/iH_V5WKQxmo a single agent (shown in purple) was trained with the social influence reward. Unlike the other agents, which continue to move and explore randomly while waiting for apples to spawn, the influencer only traverses the map when it is pursuing an apple, then stops. The rest of the time it stays still.

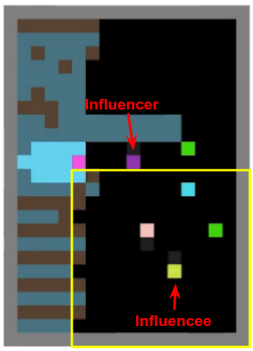


Figure 2: A moment of high influence when the purple influencer signals the presence of an apple (green tiles) outside the yellow influencee's field-of-view (yellow outlined box).

Figure 2 shows a moment of high influence between the influencer and the yellow influencee. The influencer has chosen to move towards an apple that is outside of the egocentric field-of-view of the yellow agent. Because the influencer only moves when apples are available, this signals to the yellow agent that an apple must be present above it which it cannot see. This changes the yellow agent's distribution over its planned action, $p(a_t^j|a_t^k,s_t^j)$, and allows the purple agent to gain influence. A similar moment occurs when the influencer signals to an agent that has been cleaning the river that no apples have appeared by staying still (see Figure 14 in the Supplementary Material).

In this case study, the influencer agent learned to use its own actions as a binary code which signals the presence or absence of apples in the environment. We observe a similar effect in *Harvest*. This type of action-based communication could be likened to the bee waggle dance discovered by von Frisch (1969). Evidently, the influence reward gave rise not only to cooperative behavior, but to emergent communication.

It is important to consider the limitations of the influence reward. Whether it will always give rise to cooperative behavior may depend on the specifics of the environment and task, and tuning the trade-off between environmental and influence reward. Although influence is arguably necessary for coordination (e.g. two agents coordinating to manipulate an object must have a high degree of influence between their actions), it may be possible to influence another agent in a non-cooperative way. The results provided here show that the influence reward did lead to increased cooperation, in spite of cooperation being difficult to achieve in these environments.

5. Influential Communication

Given the above results, we next experiment with using the influence reward to train agents to use an explicit communication channel. We take some inspiration from research drawing a connection between influence and communication in human learning. According to Melis & Semmann (2010), human children rapidly learn to use communication to influence the behavior of others when engaging in cooperative activities. They explain that "this ability to influence the partner via communication has been interpreted as evidence for a capacity to form shared goals with others", and that this capacity may be "what allows humans to engage in a wide range of cooperative activities".

Thus, we equip agents with an explicit communication channel, similar to the approach used by Foerster et al. (2016). At each timestep, each agent k chooses a discrete communication symbol m_t^k; these symbols are concatenated into a combined message vector $\boldsymbol{m}_t = [m_t^0, m_t^1 ... m_t^N]$, for N agents. This message vector $\boldsymbol{m}_t$ is then given as input to every other agent in the next timestep. Note that previous work has shown that self-interested agents do not learn to use this type of ungrounded, *cheap talk* communication channel effectively (Crawford & Sobel, 1982; Cao et al., 2018; Foerster et al., 2016; Lazaridou et al., 2018).

To train the agents to communicate, we augment our initial network with an additional A3C output head, that learns a communication policy π_m and value function V_m to determine which symbol to emit (see Figure 3). The normal policy and value function used for acting in the environment, π_e and V_e, are trained only with environmental reward e. We use the influence reward as an additional incentive for training the communication policy, π_m, such that $r = \alpha e + \beta c$. Counterfactuals are employed to assess how much influence an agent's communication message from the previous timestep,

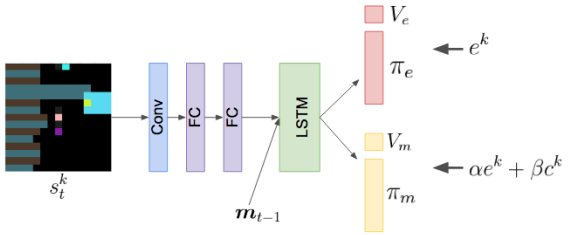
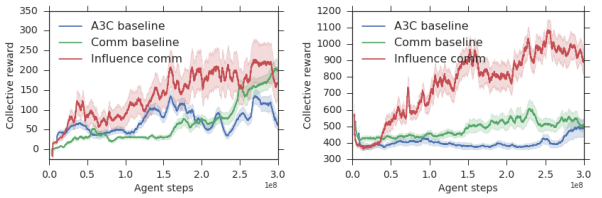


Figure 3: The communication model has two heads, which learn the environment policy, π_e, and a policy for emitting communication symbols, π_m. Other agents' communication messages $\boldsymbol{m}_{t-1}$ are input to the LSTM.



(a) *Cleanup* (b) *Harvest*

Figure 4: Total collective reward for deep RL agents with communication channels. Once again, the influence reward is essential to improve or achieve any learning.

m_{t-1}^k, has on another agent's action, a_t^j, where:

$$c_t^k = \sum_{j=0, j\neq k}^{N} \Big[D_{KL}[p(a_t^j \,|\, m_{t-1}^k, s_t^j) \Big\| p(a_t^j \,|\, s_t^j)] \Big] \qquad (2)$$

Importantly, rewarding influence through a communication channel does not suffer from the limitation mentioned in the previous section, i.e. that it may be possible to influence another agent in a non-cooperative way. We can see this for two reasons. First, there is nothing that compels agent j to act based on agent k's communication message; if m_t^k does not contain valuable information, j is free to ignore it. Second, because j's action policy π_e is trained only with environmental reward, j will only change its intended action as a result of observing m_t^k (i.e. be influenced by m_t^k) if it contains information that helps j to obtain environmental reward. Therefore, we hypothesize that influential communication must provide useful information to the listener.

5.1. Experiment II: Influential Communication

Figure 4 shows the collective reward obtained when training the agents to use an explicit communication channel. Here, the ablated model has the same structure as in Figure 3, but the communication policy π_m is trained only with environmental reward. We observe that the agents incentivized to communicate via the social influence reward learn faster, and achieve significantly higher collective reward for the majority of training in both games. In fact, in the case of *Cleanup*, we found that $\alpha = 0$ in the optimal hyperparameter setting, meaning that it was most effective to train the communication head with zero extrinsic reward (see Table 2 in the Supplementary Material). This suggests that influence alone can be a sufficient mechanism for training an effective communication policy. In *Harvest*, once again influence is critical to allow agents to learn coordinated policies and attain high reward.

To analyze the communication behaviour learned by the agents, we introduce three metrics, partially inspired by (Bogin et al., 2018). *Speaker consistency*, is a normalized score $\in [0,1]$ which assesses the entropy of $p(a^k|m^k)$ and $p(m^k|a^k)$ to determine how consistently a *speaker* agent emits a particular symbol when it takes a particular action, and vice versa (the formula is given in the Supplementary Material Section 10.4.4). We expect this measure to be high if, for example, the speaker always emits the same symbol when it is cleaning the river. We also introduce two measures of *instantaneous coordination* (IC), which are both measures of mutual information (MI): (1) symbol/action IC $= I(m_t^k; a_{t+1}^j)$ measures the MI between the influencer/speaker's symbol and the influencee/listener's next action, and (2) action/action IC $= I(a_t^k; a_{t+1}^j)$ measures the MI between the influencer's action and the influencee's next action. To compute these measures we first average over all trajectory steps, then take the maximum value between any two agents, to determine if any pair of agents are coordinating. Note that these measures are all *instantaneous*, as they consider only short-term dependencies across two consecutive timesteps, and cannot capture if an agent communicates influential compositional messages, i.e. information that requires several consecutive symbols to transmit and only then affects the other agents behavior.

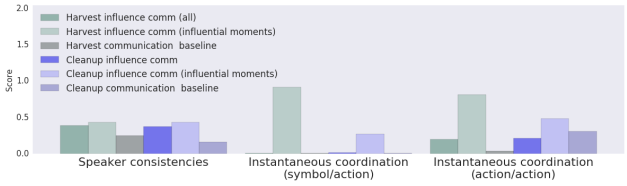


Figure 5: Metrics describing the quality of learned communication protocols. The models trained with influence reward exhibit more consistent communication and more coordination, especially in moments where influence is high.

Figure 5 presents the results. The speaker consistencies metric reveals that influence agents more unambiguously communicate about their own actions than baseline agents, indicating that the emergent communication is more meaningful. The IC metrics demonstrate that baseline agents show almost no signs of co-ordinating behavior with communication, i.e. speakers saying A and listeners doing B consistently. This result is aligned with both theoretical re-

sults in cheap-talk literature (Crawford & Sobel, 1982), and recent empirical results in MARL (e.g. Foerster et al. (2016); Lazaridou et al. (2018); Cao et al. (2018)).

In contrast, we do see high IC between influence agents, but only when we limit the analysis to timesteps on which influence was greater than or equal to the mean influence (cf. *influential moments* in Figure 5). Inspecting the results reveals a common pattern: influence is sparse in time. An agent's influence is only greater than its mean influence in less than 10% of timesteps. Because the listener agent is not compelled to listen to any given speaker, listeners selectively listen to a speaker only when it is beneficial, and influence cannot occur all the time. Only when the listener decides to change its action based on the speaker's message does influence occur, and in these moments we observe high $I(m_t^k; a_{t+1}^j)$. It appears the influencers have learned a strategy of communicating meaningful information about their own actions, and gaining influence when this becomes relevant enough for the listener to act on it.

Examining the relationship between the degree to which agents were influenced by communication and the reward they obtained gives a compelling result: agents that are the most influenced also achieve higher individual environmental reward. We sampled 100 different experimental conditions (i.e., hyper-parameters and random seeds) for both games, and normalized and correlated the influence and individual rewards. We found that agents who are more often influenced tend to achieve higher task reward in both *Cleanup*, $\rho = .67$, $p < 0.001$, and *Harvest*, $\rho = .34$, $p < 0.001$. This supports the hypothesis that in order to influence another agent via communication, the communication message should contain information that helps the listener maximize its own environmental reward. Since better listeners/influencees are more successful in terms of task reward, we have evidence that useful information was transmitted to them.

This result is promising, but may depend on the specific experimental approach taken here, in which agents interact with each other repeatedly. In this case, there is no advantage to the speaker for communicating unreliable information (i.e. lying), because it would lose influence with the listener over time. This may not be guaranteed in one-shot interactions. However, given repeated interactions, the above results provide empirical evidence that social influence as intrinsic motivation allows agents to learn meaningful communication protocols when this is otherwise not possible.

6. Modeling Other Agents

Computing the causal influence reward as introduced in Section 4 requires knowing the probability of another agent's action given a counterfactual, which we previously solved by using a centralized training approach in which agents could access other agents' policy networks. While using a centralized training framework is common in MARL (e.g. Foerster et al. (2017; 2016)), it is less realistic than a scenario in which each agent is trained independently. We can relax this assumption and achieve independent training by equipping each agent with its own internal *Model of Other Agents* (MOA). The MOA consists of a second set of fully-connected and LSTM layers connected to the agent's convolutional layer (see Figure 6), and is trained to predict all other agents' next actions given their previous actions, and the agent's egocentric view of the state: $p(\boldsymbol{a}_{t+1}|\boldsymbol{a}_t, s_t^k)$. The MOA is trained using observed action trajectories and cross-entropy loss.

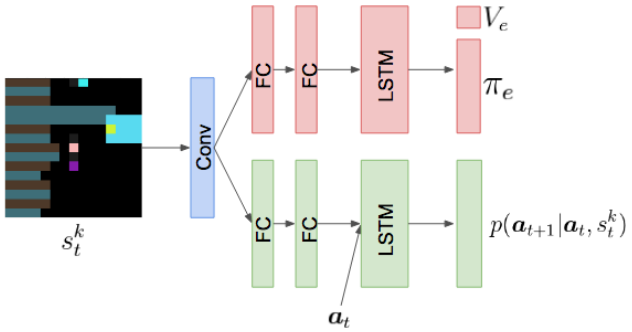


Figure 6: The Model of Other Agents (MOA) architecture learns both an RL policy π_e, and a supervised model that predicts the actions of other agents, $\boldsymbol{a}_{t+1}$. The supervised model is used for internally computing the influence reward.

A trained MOA can be used to compute the social influence reward in the following way. Each agent can "imagine" counterfactual actions that it could have taken at each timestep, and use its internal MOA to predict the effect on other agents. It can then give itself reward for taking actions that it estimates were the most influential. This has an intuitive appeal, because it resembles how humans reason about their effect on others (Ferguson et al., 2010). We often find ourselves asking counterfactual questions of the form, "How would she have acted if I had done something else in that situation?", which we answer using our internal model of others.

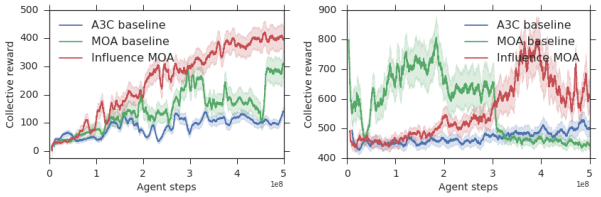
Learning a model of $p(a_{t+1}^j|a_t^k, s_t^k)$ requires implicitly modeling both other agents' internal states and behavior, as well as the environment transition function. If the model is inaccurate, this would lead to noisy estimates of the causal influence reward. To compensate for this, We only give the influence reward to an agent (k) when the agent it is attempting to influence (j) is within its field-of-view, because the estimates of $p(a_{t+1}^j|a_t^k, s_t^k)$ are more accurate when j is visible to k.[2] This constraint could have the side-effect of encouraging agents to stay in closer proximity. However, an intrinsic social reward encouraging proximity is reasonable given that humans seek affiliation and to spend time near other people (Tomasello,

[2]This contrasts with our previous models in which the influence reward was obtained even from non-visible agents.

2009).

6.1. Experiment III: Modeling Other Agents

As before, we allow the policy LSTM of each agent to condition on the actions of other agents in the last timestep (actions are visible). We compare against an ablated version of the architecture shown in Figure 6, which does not use the output of the MOA to compute a reward; rather, the MOA can be thought of as an unsupervised auxiliary task that may help the model to learn a better shared embedding layer, encouraging it to encode information relevant to predicting other agents' behavior. Figure 7 shows the collective reward obtained for agents trained with a MOA module. While we see that the auxiliary task does help to improve reward over the A3C baseline, the influence agent gets consistently higher collective reward. These results demonstrate that the influence reward can be effectively computed using an internal MOA, and thus agents can learn socially but independently, optimizing for a social reward without a centralized controller.



(a) *Cleanup* (b) *Harvest*

Figure 7: Total collective reward for MOA models. Again, intrinsic influence consistently improves learning, with the powerful A3C agent baselines not being able to learn.

Agents with influence achieve higher collective reward than the previous state-of-the-art for these environments (275 for *Cleanup* and 750 for *Harvest*) (Hughes et al., 2018). This is compelling, given that previous work relied on the assumption that agents could view one another's rewards; we make no such assumption, instead relying only on agents viewing each other's actions. Table 4 of the Supplementary Material gives the final collective reward obtained in previous work, and by each influence model for all three experiments.

7. Related work

Several attempts have been made to develop intrinsic social rewards.[3] Sequeira et al. (2011) developed hand-crafted rewards for a foraging environment, in which agents were punished for eating more than their fair share of food. Another approach gave agents an emotional intrinsic reward based on their perception of their neighbours' cooperativeness in a networked version of the iterated prisoner's dilemma, but is limited to scenarios in which it is possible to directly classify each action as cooperative or non-cooperative (Yu et al., 2013). This is untenable in complex settings with long-term strategies, such as the SSDs under investigation here.

Some approaches allow agents to view each others' rewards in order to optimize for collective reward. Peysakhovich & Lerer (2018) show that if even a single agent is trained to optimize for others' rewards, it can significantly help the group. Hughes et al. (2018) introduced an inequity aversion motivation, which penalized agents if their rewards differed too much from those of the group. Liu et al. (2014) train agents to learn their own optimal reward function in a cooperative, multi-agent setting with known group reward. However, the assumption that agents can view and optimize for each others' rewards may be unrealistic. Thus, recent work explores training agents that learn when to cooperate based solely on their own past rewards (Peysakhovich & Lerer, 2017).

Training agents to learn emergent communication protocols has been explored (Foerster et al., 2016; Cao et al., 2018; Choi et al., 2018; Lazaridou et al., 2018; Bogin et al., 2018), with many authors finding that selfish agents do not learn to use an ungrounded, *cheap talk* communication channel effectively. Crawford & Sobel (1982) find that in theory, the information communicated is proportional to the amount of common interest; thus, as agents' interests diverge, no communication is to be expected. And while communication can emerge when agents are prosocial (Foerster et al., 2016; Lazaridou et al., 2018), curious (Oudeyer & Kaplan, 2006; Oudeyer & Smith, 2016; Forestier & Oudeyer, 2017), or hand-crafted (Crandall et al., 2017), self-interested agents do not to learn to communicate (Cao et al., 2018). We have shown that the social influence reward can encourage agents to learn to communicate more effectively in complex environments.

Our MOA is related to work on machine theory of mind (Rabinowitz et al., 2018), which demonstrated that a model trained to predict agents' actions can model false beliefs. LOLA agents model the impact of their policy on the parameter updates of other agents, and directly incorporate this into the agent's own learning rule (Foerster et al., 2018).

Barton et al. (2018) propose causal influence as a way to measure coordination between agents, specifically using Convergence Cross Mapping (CCM) to analyze the degree of dependence between two agents' policies. The limitation if CCM is that estimates of causality are known to degrade in the presence of stochastic effects (Tajima et al., 2015). Counterfactual reasoning has also been used in a multi-agent setting, to marginalize out the effect of one agent on a predicted global value function estimating collective reward, and thus obtain an improved baseline for computing each agent's advantage function (Foerster et al., 2017). A similar paper shows that counterfactuals can be used with potential-based reward shaping to improve credit assignment for training a joint policy in multi-agent RL (Devlin et al., 2014). However,

[3] Note that *intrinsic* is not a synonym of *internal*; other people can be intrinsically motivating (Stavropoulos & Carver, 2013).

once again these approaches rely on a centralized controller.

Mutual information (MI) has been explored as a tool for designing social rewards. Strouse et al. (2018) train agents to optimize the MI between their actions and a categorical goal, as a way to signal or hide the agent's intentions. However, this approach depends on agents pursuing a known, categorical goal. Guckelsberger et al. (2018), in pursuit of the ultimate video game adversary, develop an agent that maximizes its empowerment, minimizes the player's empowerment, and maximizes its empowerment over the player's next state. This third goal, termed *transfer empowerment*, is obtained by maximizing the MI between the agent's actions and the player's future state. While a social form of empowerment, the authors find that agents trained with transfer empowerment simply tend to stay near the player. Further, the agents are not trained with RL, but rather analytically compute these measures in simple grid-world environments. As such, the agent cannot learn to model other agents or the environment.

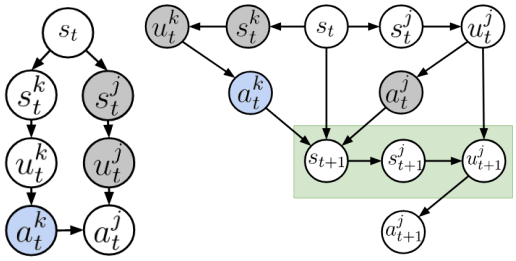
Given the social influence reward incentivizes maximizing the mutual information between agents' actions, our work also has ties to the literature on empowerment, in which agents maximize the mutual information between their actions and their future state (Klyubin et al., 2005; Mohamed & Rezende, 2015). Thus, our proposed reward can be seen as a novel social form of empowerment.

8. Details on Causal Inference

The causal influence reward presented in Eq. 4 is assessed using counterfactual reasoning. Unlike a *do*-calculus intervention (which estimates the general expected causal effect of one variable on another), a counterfactual involves conditioning on a set of variables observed in a given situation and asking how would the outcome have changed if some variable were different, and all other variables remained the same (Pearl et al., 2016). This type of inquiry allows us to measure the precise causal effect of agent k's action at timestep t, a_t^k, on agent j's action, a_t^j, in the specific environment state s_t, providing a richer and less sparse reward for agent k. Computing counterfactuals requires conditioning on the correct set of observed variables to ensure there are no confounds. In our case, the conditioning set must include not only an agent's partially observed view of the environment state, s_t^j, but also the agent's internal LSTM state u_t^j, to remove any dependency on previous timesteps in the trajectory. Thus, the basic causal influence reward can be more accurately written:

$$c_t^k = \sum_{j=0, j\neq k}^{N} \Big[D_{KL}[p(a_t^j \mid a_t^k, s_t^j, u_t^j) || p(a_t^j \mid s_t^j, u_t^j)] \Big]. \quad (3)$$

Figure 8 shows the causal diagrams for computing the influence reward in both the basic case (8a) and the MOA case (8b). Because basic influence looks at influence between agents' actions in the same timestep, the diagram is much simpler. However, to avoid circular dependencies in the graph, it requires that agent k choose its action before j, and therefore k can influence j but j cannot influence k. If there are more than two agents, we assume a disjoint set of influencer and influencee agents, and all influencers must act first.



(a) Basic (b) MOA

Figure 8: Causal diagrams of agent k's effect on j's action. Shaded nodes are conditioned on, and we intervene on a_t^k (blue node) by replacing it with counterfactuals. Nodes with a green background must be modeled using the MOA module. Note that there is no backdoor path between a_t^k and s_t in the MOA case, since it would require traversing a collider that is not in the conditioning set.

Computing influence across timesteps, as in the communication and MOA experiments, complicates the causal diagram, but ensures that each agent can influence every other agent. Figure 8b shows the diagram in the MOA case, in which we can isolate the causal effect of a_t^k on a_{t+1}^j because the back-door path through s_t is blocked by the collider nodes at s_{t+1} and u_{t+1}^j (Pearl et al., 2016). Note that it would be sufficient to condition only on s_t^k in order to block all back-door paths in this case, but we show $\langle u_t^k, s_t^k, a_t^j \rangle$ as shaded because all of these are given as inputs to the MOA to help it predict a_{t+1}^j. For the MOA to accurately estimate $p(a_{t+1}^j | a_t^k, s_t^k)$, it must model both the environment transition function T, as well as aspects of the internal LSTM state of the other agent, u_{t+1}^j, as shown by the shaded green variables in Figure 8b.

This is a simple case of counterfactual reasoning, that does not require using abduction to update the probability of any unobserved variables (Pearl, 2013). This is because we have built all relevant models, know all of their inputs, and can easily store the values for those variables at every step of the trajectory in order to condition on them so that there are no unobserved variables that could act as a confounder.

9. Conclusions and Future Work

All three experiments have shown that the proposed intrinsic social influence reward consistently leads to higher collective return. Despite variation in the tasks, hyper-parameters, neural network architectures and experimental setups, the learning curves for agents trained with the influence reward

are significantly better than the curves of powerful agents such as A3C and their improved baselines. In some cases, it is clear that influence is essential to achieve any form of learning, attesting to the promise of this idea and highlighting the complexity of learning general deep neural network multi-agent policies.

Experiment I also showed that the influence reward can lead to the emergence of communication protocols. In experiment II, which included an explicit communication channel, we saw that influence improved communication. Experiment III showed that influence can be computed by augmenting agents with an internal model of other agents. The influence reward can thus be computed without having access to another agent's reward function, or requiring a centralized controller. We were able to surpass state-of-the-art performance on the SSDs studied here, despite the fact that previous work relied on agents' ability to view other agents' rewards.

Using counterfactuals to allow agents to understand the effects of their actions on others is a promising approach with many extensions. Agents could use counterfactuals to develop a form of 'empathy', by simulating how their actions affect another agent's value function. Influence could also be used to drive coordinated behavior in robots attempting to do cooperative manipulation and control tasks. Finally, if we view multi-agent networks as single agents, influence could be used as a regularizer to encourage different modules of the network to integrate information from other networks; for example, to hopefully prevent collapse in hierarchical RL.

Acknowledgements

We are grateful to Eugene Vinitsky for his help in reproducing the SSD environments in open source to improve the replicability of the paper. We also thank Steven Wheelwright, Neil Rabinowitz, Thore Graepel, Alexander Novikov, Scott Reed, Pedro Mediano, Jane Wang, Max Kleiman-Weiner, Andrea Tacchetti, Kevin McKee, Yannick Schroecker, Matthias Bauer, David Rolnick, Francis Song, David Budden, and Csaba Szepesvari, as well as everyone on the DeepMind Machine Learning and Multi-Agent teams for their helpful discussions and support.

References

Barton, S. L., Waytowich, N. R., Zaroukian, E., and Asher, D. E. Measuring collaborative emergent behavior in multi-agent reinforcement learning. *arXiv preprint arXiv:1807.08663*, 2018.

Bogin, B., Geva, M., and Berant, J. Emergence of communication in an interactive world with consistent speakers. *arXiv preprint arXiv:1809.00549*, 2018.

Cao, K., Lazaridou, A., Lanctot, M., Leibo, J. Z., Tuyls, K., and Clark, S. Emergent communication through negotiation. *arXiv preprint arXiv:1804.03980*, 2018.

Capdepuy, P., Polani, D., and Nehaniv, C. L. Maximization of potential information flow as a universal utility for collective behaviour. In *Artificial Life, 2007. ALIFE'07. IEEE Symposium on*, pp. 207–213. Ieee, 2007.

Choi, E., Lazaridou, A., and de Freitas, N. Compositional obverter communication learning from raw visual input. *arXiv preprint arXiv:1804.02341*, 2018.

Crandall, J. W., Oudah, M., Chenlinangjia, T., Ishowo-Oloko, F., Abdallah, S., Bonnefon, J., Cebrián, M., Shariff, A., Goodrich, M. A., and Rahwan, I. Cooperating with machines. *CoRR*, abs/1703.06207, 2017. URL http://arxiv.org/abs/1703.06207.

Crawford, V. P. and Sobel, J. Strategic information transmission. *Econometrica: Journal of the Econometric Society*, pp. 1431–1451, 1982.

Devlin, S., Yliniemi, L., Kudenko, D., and Tumer, K. Potential-based difference rewards for multiagent reinforcement learning. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pp. 165–172. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

Ferguson, H. J., Scheepers, C., and Sanford, A. J. Expectations in counterfactual and theory of mind reasoning. *Language and Cognitive Processes*, 25(3):297–346, 2010. doi: 10.1080/01690960903041174. URL https://doi.org/10.1080/01690960903041174.

Foerster, J., Assael, I. A., de Freitas, N., and Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 2137–2145, 2016.

Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. *arXiv preprint arXiv:1705.08926*, 2017.

Foerster, J., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., and Mordatch, I. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 122–130. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Forestier, S. and Oudeyer, P.-Y. A unified model of speech and tool use early development. In *39th Annual Conference of the Cognitive Science Society (CogSci 2017)*, 2017.

Gers, F. A., Schmidhuber, J., and Cummins, F. Learning to forget: Continual prediction with lstm. 1999.

Guckelsberger, C., Salge, C., and Togelius, J. New and surprising ways to be mean. adversarial npcs with coupled empowerment minimisation. *arXiv preprint arXiv:1806.01387*, 2018.

Harari, Y. N. *Sapiens: A brief history of humankind*. Random House, 2014.

Henrich, J. *The Secret of Our Success: How culture is driving human evolution, domesticating our species, and making us smart*. Princeton University Press, Princeton, NJ, 2015. URL http://press.princeton.edu/titles/10543.html.

Herrmann, E., Call, J., Hernàndez-Lloreda, M. V., Hare, B., and Tomasello, M. Humans have evolved specialized skills of social cognition: The cultural intelligence hypothesis. *Science*, 317(5843):1360–1366, 2007. ISSN 0036-8075. doi: 10.1126/science.1146282. URL http://science.sciencemag.org/content/317/5843/1360.

Hughes, E., Leibo, J. Z., Phillips, M. G., Tuyls, K., Duéñez-Guzmán, E. A., Castañeda, A. G., Dunning, I., Zhu, T., McKee, K. R., Koster, R., et al. Inequity aversion improves cooperation in intertemporal social dilemmas. In *Advances in neural information processing systems (NIPS)*, Montreal, Canada, 2018.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. Empowerment: A universal agent-centric measure of control. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 1, pp. 128–135. IEEE, 2005.

Laland, K. N. *Darwin's unfinished symphony : how culture made the human mind / Kevin N. Laland*. Princeton University Press Princeton, 2017. ISBN 9781400884872 140088487.

Lazaridou, A., Hermann, K. M., Tuyls, K., and Clark, S. Emergence of linguistic communication from referential games with symbolic and pixel input. *arXiv preprint arXiv:1804.03984*, 2018.

Leibo, J. Z., Zambaldi, V., Lanctot, M., Marecki, J., and Graepel, T. Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pp. 464–473. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

Liu, B., Singh, S., Lewis, R. L., and Qin, S. Optimal rewards for cooperative agents. *IEEE Transactions on Autonomous Mental Development*, 6(4):286–297, 2014.

Lizier, J. T. and Prokopenko, M. Differentiating information transfer and causal effect. *The European Physical Journal B*, 73(4):605–615, 2010.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, O. P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pp. 6379–6390, 2017.

Melis, A. P. and Semmann, D. How is human cooperation different? *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 365(1553):2663–2674, 2010.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937, 2016.

Mohamed, S. and Rezende, D. J. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pp. 2125–2133, 2015.

Oudeyer, P.-Y. and Kaplan, F. Discovering communication. *Connection Science*, 18(2):189–206, 2006.

Oudeyer, P.-Y. and Smith, L. B. How evolution may work through curiosity-driven developmental process. *Topics in Cognitive Science*, 8(2):492–502, 2016.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, volume 2017, 2017.

Pearl, J. Structural counterfactuals: A brief introduction. *Cognitive science*, 37(6):977–985, 2013.

Pearl, J., Glymour, M., and Jewell, N. P. *Causal inference in statistics: a primer*. John Wiley & Sons, 2016.

Perolat, J., Leibo, J. Z., Zambaldi, V., Beattie, C., Tuyls, K., and Graepel, T. A multi-agent reinforcement learning model of common-pool resource appropriation. In *Advances in Neural Information Processing Systems*, pp. 3643–3652, 2017.

Peysakhovich, A. and Lerer, A. Consequentialist conditional cooperation in social dilemmas with imperfect information. *arXiv preprint arXiv:1710.06975*, 2017.

Peysakhovich, A. and Lerer, A. Prosocial learning agents solve generalized stag hunts better than selfish ones. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2043–2044. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Rabinowitz, N. C., Perbet, F., Song, H. F., Zhang, C., Eslami, S., and Botvinick, M. Machine theory of mind. *arXiv preprint arXiv:1802.07740*, 2018.

Schelling, T. C. Hockey helmets, concealed weapons, and daylight saving: A study of binary choices with externalities. *Journal of Conflict resolution*, 17(3):381–428, 1973.

Schmidhuber, J. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

Sequeira, P., Melo, F. S., Prada, R., and Paiva, A. Emerging social awareness: Exploring intrinsic motivation in multiagent learning. In *Development and Learning (ICDL), 2011 IEEE International Conference on*, volume 2, pp. 1–6. IEEE, 2011.

Singh, S. P., Barto, A. G., and Chentanez, N. Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pp. 1281–1288, 2004.

Stavropoulos, K. K. and Carver, L. J. Research review: social motivation and oxytocin in autism–implications for joint attention development and intervention. *Journal of Child Psychology and Psychiatry*, 54(6):603–618, 2013.

Strouse, D., Kleiman-Weiner, M., Tenenbaum, J., Botvinick, M., and Schwab, D. Learning to share and hide intentions using information regularization. *arXiv preprint arXiv:1808.02093*, 2018.

Tajima, S., Yanagawa, T., Fujii, N., and Toyoizumi, T. Untangling brain-wide dynamics in consciousness by cross-embedding. *PLoS computational biology*, 11(11): e1004537, 2015.

Tomasello, M. *Why we cooperate*. MIT press, 2009.

van Schaik, C. P. and Burkart, J. M. Social learning and evolution: the cultural intelligence hypothesis. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1567):1008–1016, 2011.

von Frisch, K. The dance language and orientation of bees. 5, 06 1969.

Yu, C., Zhang, M., and Ren, F. Emotional multiagent reinforcement learning in social dilemmas. In *International Conference on Principles and Practice of Multi-Agent Systems*, pp. 372–387. Springer, 2013.

10. Supplementary Material

10.1. Influence as Mutual Information

The causal influence of agent k on agent j is:

$$D_{KL}\Big[p(a_t^j \mid a_t^k, z_t)\Big\|p(a_t^j \mid z_t)\Big], \tag{4}$$

where z_t represents all relevant u and s background variables at timestep t. The influence reward to the mutual information (MI) between the actions of agents k and j, which is given by

$$\begin{aligned} I(A^j;A^k|z) &= \sum_{a^k,a^j} p(a^j,a^k|z)\log\frac{p(a^j,a^k|z)}{p(a^j|z)p(a^k|z)} \\ &= \sum_{a^k} p(a^k|z) D_{\text{KL}}\Big[p(a^j|a^k,z)\Big\|p(a^j|z)\Big], \end{aligned} \tag{5}$$

where we see that the D_{KL} factor in Eq. 5 is the causal influence reward given in Eq. 4.

By sampling N independent trajectories τ_n from the environment, where k's actions a_n^k are drawn according to $p(a^k|z)$, we perform a Monte-Carlo approximation of the MI (see e.g. Strouse et al. (2018)),

$$\begin{aligned} I(A^k;A^j|z) &= \mathbb{E}_\tau\Big[D_{\text{KL}}\big[p(A^j|A^k,z)\big\|p(A^j|z)\big]\Big|z\Big] \\ &\approx \frac{1}{N}\sum_n D_{\text{KL}}\big[p(A^j|a_n^k,z)\big\|p(A^j|z)\big]. \end{aligned} \tag{6}$$

Thus, in expectation, the social influence reward is the MI between agents' actions.

Whether the policy trained with Eq. 4 actually learns to approximate the MI depends on the learning dynamics. We calculate the intrinsic social influence reward using Eq. 4, because unlike Eq. 5, which gives an estimate of the symmetric bandwidth between k and j, Eq. 4 gives the directed causal effect of the specific action taken by agent k, a_t^k. We believe this will result in an easier reward to learn, since it allows for better credit assignment; agent k can more easily learn which of its actions lead to high influence.

The connection to mutual information is interesting, because a frequently used intrinsic motivation for single agent RL is *empowerment*, which rewards the agent for having high mutual information between its actions and the future state of the environment (e.g. Klyubin et al. (2005); Capdepuy et al. (2007)). To the extent that the social influence reward approximates the MI, k is rewarded for having empowerment over j's actions.

The social influence reward can also be computed using other divergence measures besides KL-divergence. Lizier & Prokopenko (2010) propose *local information flow* as a measure of direct causal effect; this is equivalent to the *pointwise mutual information* (the innermost term of Eq. 6), given by:

$$\begin{aligned} pmi(a^k;a^j \mid Z=z) &= \log\frac{p(a^j \mid a^k,z)}{p(a^j \mid z)} \\ &= \log\frac{p(a^k,a^j \mid z)}{p(a^k \mid z)p(a^j \mid z)}. \end{aligned} \tag{7}$$

The PMI gives us a measure of influence of a single action of k on the single action taken by j. The expectation of the PMI

over $p(a^j, a^k|z)$ is the MI. We experiment with using the PMI and a number of divergence measures, including the Jensen-Shannon Divergence (JSD), and find that the influence reward is robust to the choice of measure.

10.2. Sequential Social Dilemmas

Figure 9 depicts the SSD games under investigation. In each of the games, an agent is rewarded $+1$ for every apple it collects, but the apples are a limited resource. Agents have the ability to punish each other with a *fining beam*, which costs -1 reward to fire, and fines any agent it hits -50 reward.

In *Cleanup* (a public goods game) agents must clean a river before apples can grow, but are not able to harvest apples while cleaning. In *Harvest* (a common pool resource game), apples respawn at a rate proportional to the amount of nearby apples; if apples are harvested too quickly, they will not grow back. Both coordination, and cooperation are required to solve both games. In *Cleanup*, agents must efficiently time harvesting apples and cleaning the river, and allow agents cleaning the river a chance to consume apples. In *Harvest*, agents must spatially distribute their harvesting, and abstain from consuming apples too quickly in order to harvest sustainably. The code for these games, including hyperparameter settings and apple and waste respawn probabilities, can be found at `https://github.com/eugenevinitsky/sequential_social_dilemma_games`.

The reward structure of the games is shown in Figure 10, which gives the Schelling diagram for both SSD tasks under investigation. A Schelling diagram (Schelling, 1973; Perolat et al., 2017) depicts the relative payoffs for a single agent's strategy given a fixed number of other agents who are cooperative. These diagrams show that all agents would benefit from learning to cooperate, because even the agents that are being exploited get higher reward than in the regime where all agents defect. However, traditional RL agents struggle to learn to cooperate and solve these tasks effectively (Hughes et al., 2018).

10.3. Additional experiment - Box Trapped

As a proof-of-concept experiment to test whether the influence reward works as expected, we constructed a special environment, shown in Figure 11. In this environment, one agent (teal) is trapped in a box. The other agent (purple) has a special action it can use to open the box... or it can simply choose to consume apples, which exist outside the box and are inexhaustible in this environment.

As expected, a vanilla A3C agent learns to act selfishly; the purple agent will simply consume apples, and chooses the *open box* action in 0% of trajectories once the policy has converged. A video of A3C agents trained in this environment is available at: `https://youtu.be/C8SE9_YKzxI`, which shows that the purple agent leaves its compatriot trapped in the box throughout the trajectory.

In contrast, an agent trained with the social influence reward chooses the *open box* action in 88% of trajectories, releasing its fellow agent so that they are both able to consume apples. A video of this behavior is shown at: `https://youtu.be/Gfo248-qt3c`. Further, as Figure 12 reveals, the purple influencer agent usually chooses to open the box within the first few steps of the trajetory, giving its fellow agent more time to collect reward.

Most importantly though, Figure 13 shows the influence reward over the course of a trajectory in the *Box trapped* environment. The agent chooses the *open box* action in the second timestep; at this point, we see a corresponding spike in the influence reward. This reveals that the influence reward works as expected, incentivizing an action which has a strong — and in this case, prosocial — effect on the other agent's behavior.

10.4. Implementation details

All models are trained with a single convolutional layer with a kernel of size 3, stride of size 1, and 6 output channels. This is connected to two fully connected layers of size 32 each, and an LSTM with 128 cells. We use a discount factor $\gamma = .99$. The number of agents N is fixed to 5.

In addition to the comparison function used to compute influence (e.g. KL-divergence, PMI, JSD), there are many other hyperparameters that can be tuned for each model. We use a random search over hyperparameters, ensuring a fair comparison with the search size over the baseline parameters that are shared with the influence models. For all models we search for the optimal entropy reward and learning rate, where we anneal the learning rate from an initial value `lr_init` to `lr_final`. The below sections give the parameters found to be most effective for each of the three experiments.

10.4.1. Basic influence hyperparameters

In this setting we vary the number of influencers from $1-4$, the influence reward weight β, and the number of curriculum steps over which the weight of the influence reward is linearly increased C. In this setting, since we have a centralised controller, we also experiment with giving the influence reward to the agent being influenced as well, and find that this sometimes helps. This 'influencee' reward is not used in the other two experiments, since it precludes independent training. The hyperparameters found to give the best performance for each model are shown in Table 1.

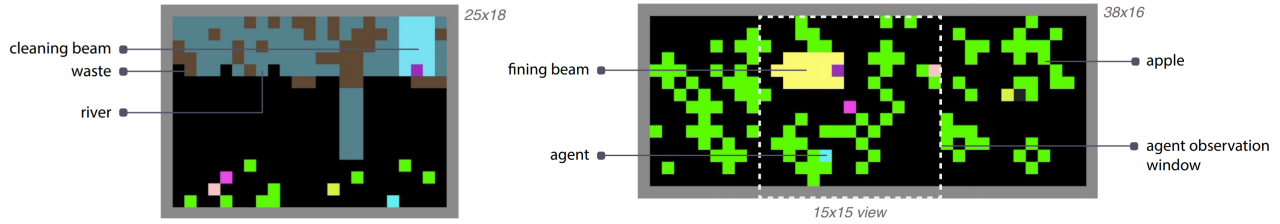


Figure 9: The two SSD environments, *Cleanup* (left) and *Harvest* (right). Agents can exploit other agents for immediate payoff, but at the expense of the long-term collective reward of the group. Reproduced with permission from Hughes et al. (2018).

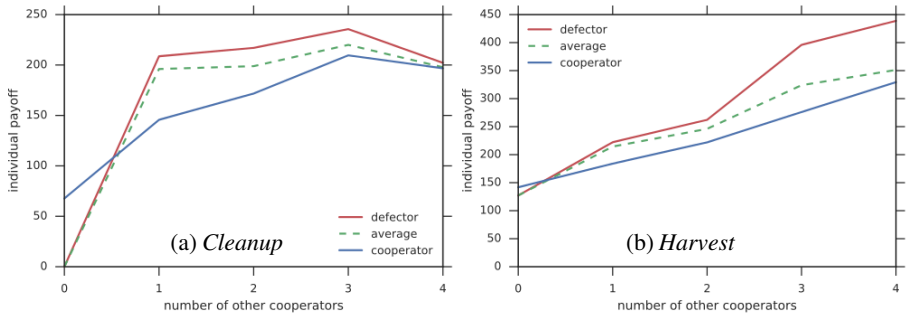


Figure 10: Schelling diagrams for the two social dilemma tasks show that an individual agent is motivated to defect, though everyone benefits when more agents cooperate. Reproduced with permission from Hughes et al. (2018).

Hyperparameter	Cleanup			Harvest		
	A3C baseline	Visible actions baseline	Influence	A3C baseline	Visible actions baseline	Influence
Entropy reg.	.00176	.00176	.000248	.000687	.00184	.00025
lr_init	.00126	.00126	.00107	.00136	.00215	.00107
lr_end	.000012	.000012	.000042	.000028	.000013	.000042
Number of influencers	-	3	1	-	3	3
Influence weight β	-	0	.146	-	0	.224
Curriculum C	-	-	140	-	-	140
Policy comparison	-	-	JSD	-	-	PMI
Influencee reward	-	-	1	-	-	0

Table 1: Optimal hyperparameter settings for the models in the basic influence experiment.

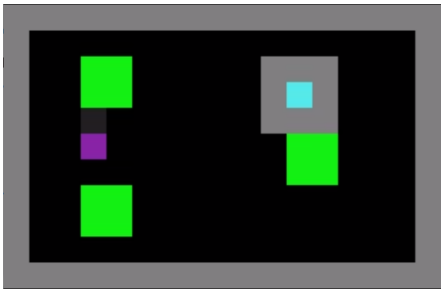


Figure 11: The *Box trapped* environment in which the teal agent is trapped, and the purple agent can release it with a special *open box* action.

10.4.2. Communication hyperparameters

Because the communication models have an extra A2C output head for the communication policy, we use an additional entropy regularization term just for this head, and apply a weight to the communication loss in the loss function. We also vary the number of communication symbols that the agents can emit, and the size of the linear layer that connects the LSTM to the communication policy layer, which we term the communication embedding size. Finally, in the communication regime, we experiment to setting the weight on the extrinsic reward E, α, to zero. The best hyperparameters for each of the communication models are shown in Table 2.

10.4.3. Model of other agents (MOA) hyperparameters

The MOA hyperparameters include whether to only train the MOA with cross-entropy loss on the actions of agents that are visible, and how much to weight the supervised loss in the overall loss of the model. The best hyperparameters are shown in Table 3.

Hyperparameter	Cleanup			Harvest		
	A3C baseline	Comm. baseline	Influence comm.	A3C baseline	Comm. baseline	Influence comm.
Entropy reg.	.00176	.000249	.00305	.000687	.000174	.00220
lr_init	.00126	.00223	.00249	.00136	.00137	.000413
lr_end	.000012	.000022	.0000127	.000028	.0000127	.000049
Influence weight β	-	0	2.752	-	0	4.825
Extrinsic reward weight α	-	-	0	-	-	1.0
Curriculum C	-	-	1	-	-	8
Policy comparison	-	-	KL	-	-	KL
Comm. entropy reg.	-	-	.000789	-	-	.00208
Comm. loss weight	-	-	.0758	-	-	.0709
Symbol vocab size	-	-	9	-	-	7
Comm. embedding	-	-	32	-	-	16

Table 2: Optimal hyperparameter settings for the models in the communication experiment.

Hyperparameter	Cleanup			Harvest		
	A3C baseline	MOA baseline	Influence MOA	A3C baseline	MOA baseline	Influence MOA
Entropy reg.	.00176	.00176	.00176	.000687	.00495	.00223
lr_init	.00126	.00123	.00123	.00136	.00206	.00120
lr_end	.000012	.000012	.000012	.000028	.000022	.000044
Influence weight β	-	0	.620	-	0	2.521
MOA loss weight	-	1.312	15.007	-	1.711	10.911
Curriculum C	-	-	40	-	-	226
Policy comparison	-	-	KL	-	-	KL
Train MOA only when visible	-	False	True	-	False	True

Table 3: Optimal hyperparameter settings for the models in the model of other agents (MOA) experiment.

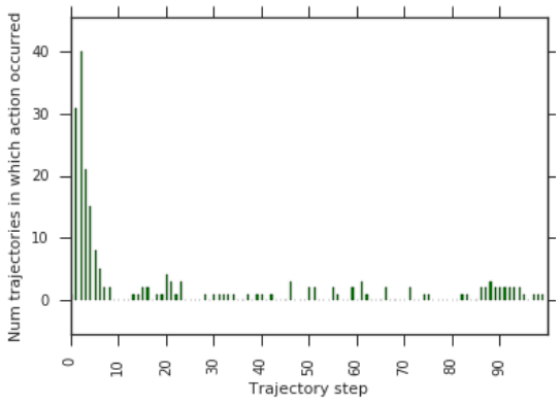


Figure 12: Number of times the *open box* action occurs at each trajectory step over 100 trajectories.

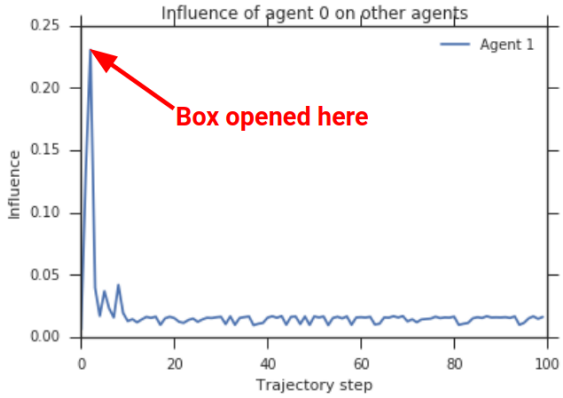


Figure 13: Influence reward over a trajectory in *Box trapped*. An agent gets high influence for letting another agent out of the box in which it is trapped.

10.4.4. Communication analysis

The speaker consistency metric is calculated as:

$$\sum_{k=1}^{N} 0.5[\sum_{c} 1 - \frac{H(p(a^k|m^k=c))}{H_{max}} + \sum_{a} 1 - \frac{H(p(m^k|a^k=a))}{H_{max}}], \quad (8)$$

where H is the entropy function and H_{max} is the maximum entropy based on the number of discrete symbols or actions. The goal of the metric is to measure how much of a 1:1 correspondence exists between a speaker's action and the speaker's communication message.

10.5. Additional results

10.5.1. Basic influence emergent communication

Figure 14 shows an additional moment of high influence in the *Cleanup* game. The purple influencer agent can see the area within the white box, and therefore all of the apple patch. The field-of-view of the magenta influencee is outlined with the magenta box; it cannot see if apples have appeared, even though it has been cleaning the river, which is the action required to cause apples to appear. When the purple influencer turns left and does not move towards the apple patch, this signals to the magenta agent that no apples have appeared, since otherwise the influence would move right.

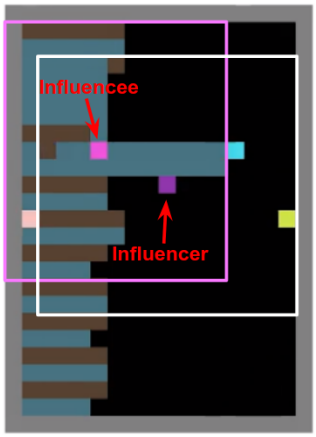
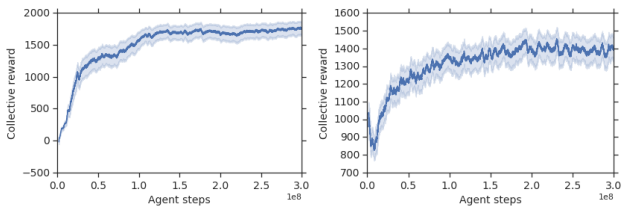


Figure 14: A moment of high influence between the purple influencer and magenta influencee.

10.5.2. Optimizing for collective reward



(a) *Cleanup* (b) *Tragedy*

Figure 15: Total collective reward obtained by agents trained to optimize for the collective reward, for the 5 best hyperparameter settings with 5 random seeds each. Error bars show a 99.5% confidence interval (CI) computed within a sliding window of 200 agent steps.

In this section we include the results of training explicitly prosocial agents, which directly optimize for the collective reward of all agents. Previous work (e.g. Peysakhovich & Lerer (2018)) has shown that training agents to optimize for the rewards of other agents can help the group to obtain better collective outcomes. Following a similar principle, we implemented agents that optimize for a convex combination of their own individual reward e_t^k and the collective reward of all other agents, $\sum_{i=1,i\neq k}^{N} e_t^i$. Thus, the reward function for agent k is $r_t^k = e_t^k + \eta \sum_{i=1,i\neq k}^{N} e_t^i$. We conducted the same hyperparameter search over the parameters mentioned in Section 10.4.1 varying the weight placed on the collective reward, $\eta \in [0,2]$.

As expected, we find that agents trained to optimize for collective reward attain higher collective reward in both *Cleanup* and *Harvest*, as is shown in Figure 15. In both games, the optimal value for $\eta = 0.85$. Interestingly, however, the equality in the individual returns for these agents is extremely low. Across the hyperparameter sweep, no solution to the *Cleanup* game which scored more than 20 points in terms of collective return was found in which all agents scored an individual return above 0. It seems that in *Cleanup*, when agents are trained to optimize for collective return, they converge on a solution in which some agents never receive any reward.

Note that training agents to optimize for collective reward requires that each agent can view the rewards obtained by other agents. As discussed previously, the social influence reward is a novel way to obtain cooperative behavior, that does not require making this assumption.

10.5.3. Collective reward and equality

It is important to note that collective reward is not always the perfect metric of cooperative behavior, a finding that was also discovered by Barton et al. (2018) and emphasized by Leibo et al. (2017). In the case, we find that there is a spurious solution to the *Harvest* game, in which one agent fails to learn and fails to collect any apples. This leads to very high collective reward, since it means there is one fewer agent that can exploit the others, and makes sustainable harvesting easier to achieve. Therefore, for the results shown in the paper, we eliminate any random seed in *Harvest* for which one of the agents has failed to learn to collect apples, as in previous work (Hughes et al., 2018).

However, here we also present an alternative strategy for assessing the overall collective outcomes: weighting the total collective reward by an index of equality of the individual rewards. Specifically, we compute the Gini coefficient over the N agents' individual environmental rewards e_t^k:

$$G = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} |e_t^i - e_t^j|}{2N \sum_{i=1}^{N} e_t^i}, \tag{9}$$

which gives us a measure of the inequality of the returns, where $G \in [0,1]$, with $G = 0$ indicating perfect equality. Thus, $1 - G$ is a measure of equality; we use this to weight the collective reward for each experiment, and plot the results in Figure 16. Once again, we see that the influence models give the highest final performance, even with this new metric.

10.5.4. COLLECTIVE REWARD OVER MULTIPLE HYPERPARAMETERS

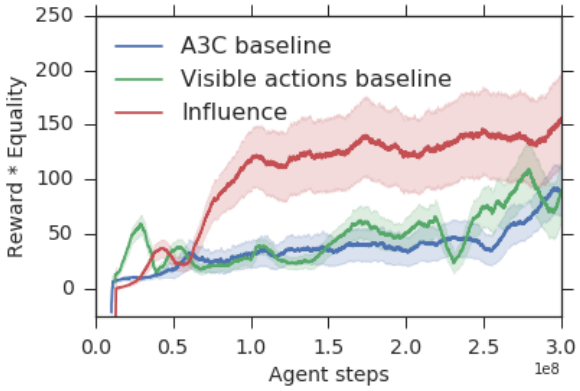
Finally, we would like to show that the influence reward is robust to the choice of hyperparameter settings. Therefore, in Figure 17, we plot the collective reward of the top 5 best hyperparameter settings for each experiment, over 5 random seeds each. Once again, the influence models result in higher collective reward, which provides evidence that the model is robust to the choice of hyperparameters.

10.5.5. PERFORMANCE COMPARISON BETWEEN MODELS AND RELATED WORK

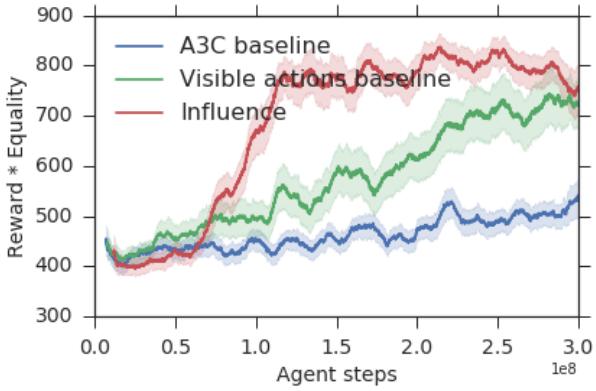
Table 4 presents the final collective reward obtained by each of the models tested in the experiments presented in the paper. We see that in several cases, the influence agents are even able to out-perform the state-of-the-art results on these tasks reported by (Hughes et al., 2018), despite the fact that the solution proposed by (Hughes et al., 2018) requires that agents can view other agents' rewards, whereas we do not make this assumption, and instead only require that agents can view each others' actions.

	Cleanup	Harvest
A3C baseline	89	485
Inequity aversion (Hughes et al.)	275	750
Influence - Basic	190	**1073**
Influence - Communication	166	**951**
Influence - Model of other agents	**392**	588

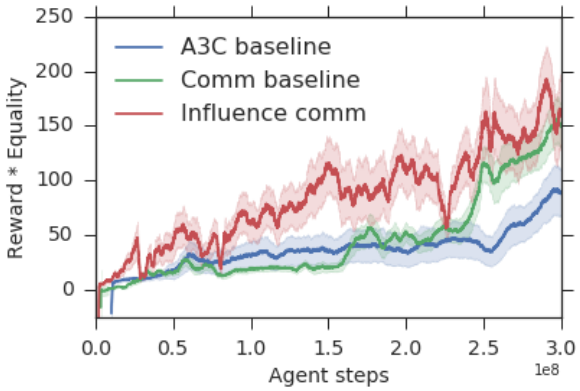
Table 4: Final collective reward over the last 50 agent steps for each of the models considered. Bolded entries represent experiments in which the influence models significantly outperformed the scores reported in previous work on *inequity aversion* (Hughes et al., 2018). This is impressive, considering the *inequity averse* agents are able to view all other agents' rewards. We make no such assumption, and yet are able to achieve similar or superior performance.



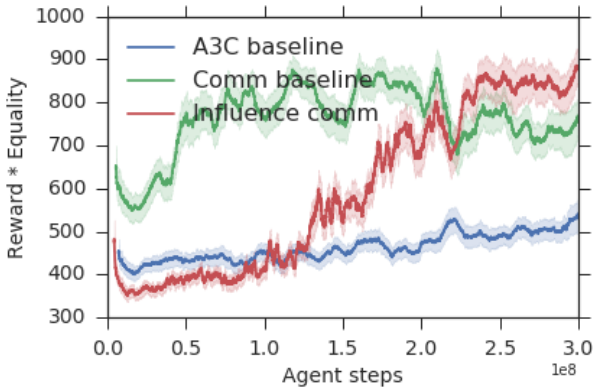
(a) *Cleanup* - Basic influence



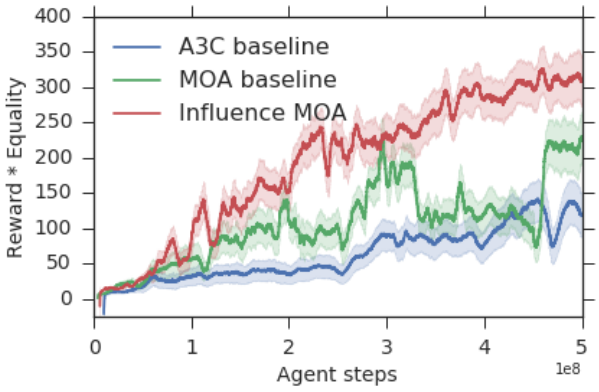
(b) *Harvest* - Basic influence



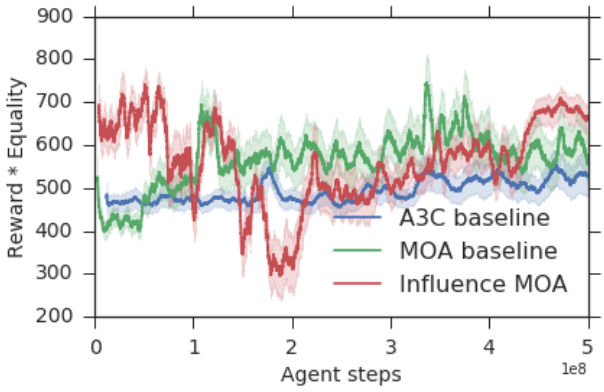
(c) *Cleanup* - Communication



(d) *Harvest* - Communication

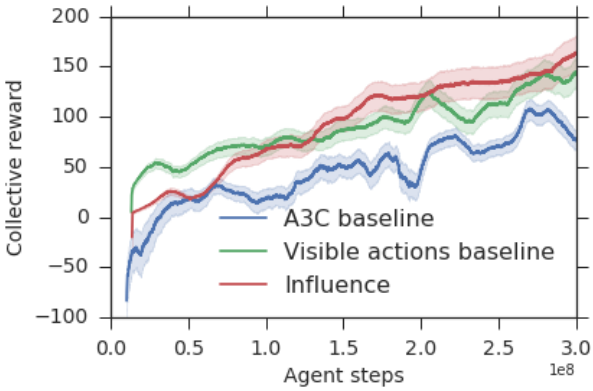


(e) *Cleanup* - Model of other agents

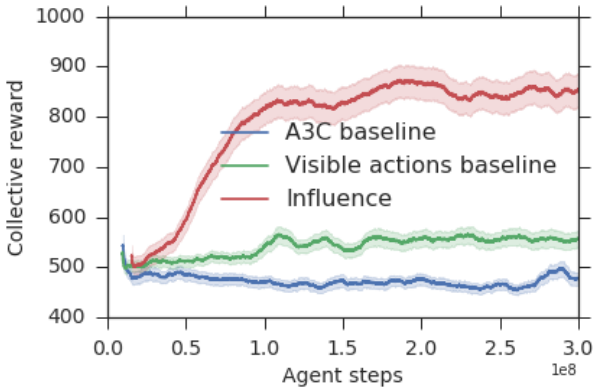


(f) *Harvest* - Model of other agents

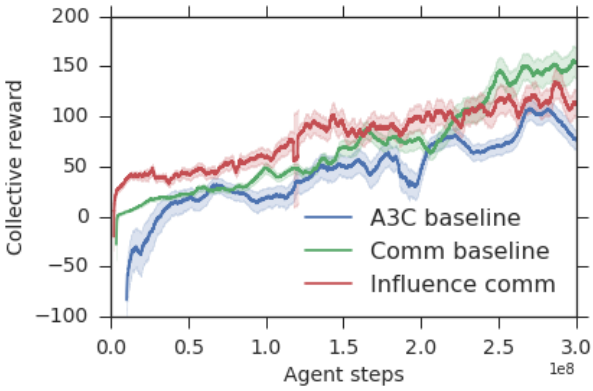
Figure 16: Total collective reward times equality, $R*(1-G)$, obtained in all experiments. Error bars show a 99.5% confidence interval (CI) over 5 random seeds, computed within a sliding window of 200 agent steps. Once again, the models trained with influence reward (red) significantly outperform the baseline and ablated models.



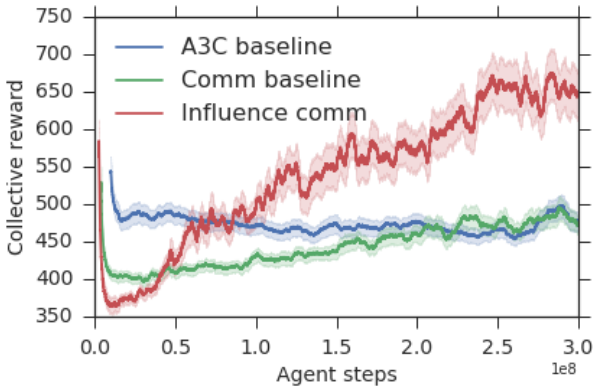
(a) *Cleanup* - Basic influence



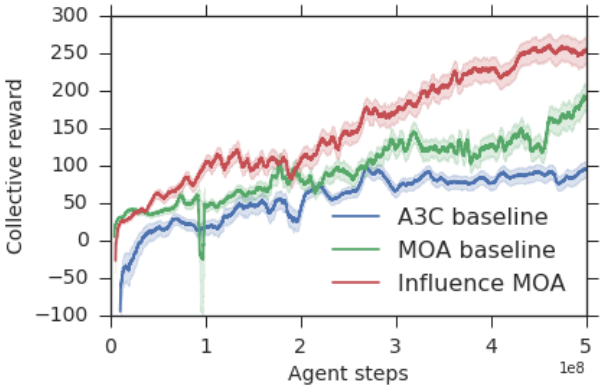
(b) *Harvest* - Basic influence



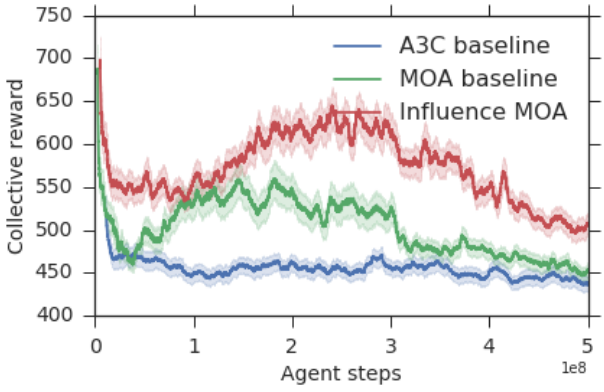
(c) *Cleanup* - Communication



(d) *Harvest* - Communication



(e) *Cleanup* - Model of other agents



(f) *Harvest* - Model of other agents

Figure 17: Total collective reward over the top 5 hyperparameter settings, with 5 random seeds each, for all experiments. Error bars show a 99.5% confidence interval (CI) computed within a sliding window of 200 agent steps. The influence models still maintain an advantage over the baselines and ablated models, suggesting the technique is robust to the hyperparameter settings.

Supplementary Material for Social Influence as Intrinsic Motivation for Multi-Agent Deep Reinforcement Learning

Anonymous Authors[1]

1. Influence as Mutual Information

The causal influence of agent k on agent j is:

$$D_{KL}\left[p(a_t^j \mid a_t^k, z_t) \middle\| p(a_t^j \mid z_t)\right], \tag{1}$$

where z_t represents the conditioning variables at timestep t, $z_t = \langle u_t^j, s_t^j \rangle$. The influence reward to the mutual information (MI) between the actions of agents k and j, which is given by

$$\begin{aligned} I(A^j; A^k|z) &= \sum_{a^k, a^j} p(a^j, a^k|z) \log \frac{p(a^j, a^k|z)}{p(a^j|z)p(a^k|z)} \\ &= \sum_{a^k} p(a^k|z) D_{\text{KL}}\left[p(a^j|a^k, z) \middle\| p(a^j|z)\right], \end{aligned} \tag{2}$$

where we see that the D_{KL} factor in Eq. 2 is the causal influence reward given in Eq. 1.

By sampling N independent trajectories τ_n from the environment, where k's actions a_n^k are drawn according to $p(a^k|z)$, we perform a Monte-Carlo approximation of the MI (see e.g. **?**),

$$\begin{aligned} I(A^k; A^j|z) &= \mathbb{E}_\tau\left[D_{\text{KL}}\left[p(A^j|A^k, z) \middle\| p(A^j|z)\right] \middle| z\right] \\ &\approx \frac{1}{N} \sum_n D_{\text{KL}}\left[p(A^j|a_n^k, z) \middle\| p(A^j|z)\right]. \end{aligned} \tag{3}$$

Thus, in expectation, the social influence reward is the MI between agents' actions.

Whether the policy trained with Eq. 1 actually learns to approximate the MI depends on the learning dynamics. We calculate the intrinsic social influence reward using Eq. 1, because unlike Eq. 2, which gives an estimate of the symmetric bandwidth between k and j, Eq. 1 gives the directed causal effect of the specific action taken by agent k, a_t^k. We believe this will result in an easier reward to learn, since it allows for better credit assignment; agent k can more easily learn which of its actions lead to high influence.

The connection to mutual information is interesting, because a frequently used intrinsic motivation for single agent RL is *empowerment*, which rewards the agent for having high mutual information between its actions and the future state of the environment (e.g. **??**). To the extent that the social influence reward approximates the MI, k is rewarded for having empowerment over j's actions.

The social influence reward can also be computed using other divergence measures besides KL-divergence. **?** propose *local information flow* as a measure of direct causal effect; this is equivalent to the *pointwise mutual information* (the innermost term of Eq. 3), given by:

$$\begin{aligned} pmi(a^k; a^j \mid Z = z) &= \log \frac{p(a^j \mid a^k, z)}{p(a^j \mid z)} \\ &= \log \frac{p(a^k, a^j \mid z)}{p(a^k \mid z)p(a^j \mid z)}. \end{aligned} \tag{4}$$

The PMI gives us a measure of influence of a single action of k on the single action taken by j. The expectation of the PMI over $p(a^j, a^k|z)$ is the MI. We experiment with using the PMI and a number of divergence measures, including the Jensen-Shannon Divergence (JSD), and find that the influence reward is robust to the choice of measure.

2. Sequential Social Dilemmas

Figure 1 depicts the SSD games under investigation. In each of the games, an agent is rewarded $+1$ for every apple it collects, but the apples are a limited resource. Agents have the ability to punish each other with a *fining beam*, which costs -1 reward to fire, and fines any agent it hits -50 reward.

In *Cleanup* (a public goods game) agents must clean a river before apples can grow, but are not able to harvest apples while cleaning. In *Harvest* (a common pool resource game), apples respawn at a rate proportional to the amount of nearby apples; if apples are harvested too quickly, they will not grow back. Both coordination, and cooperation are required to solve both games. In *Cleanup*, agents

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

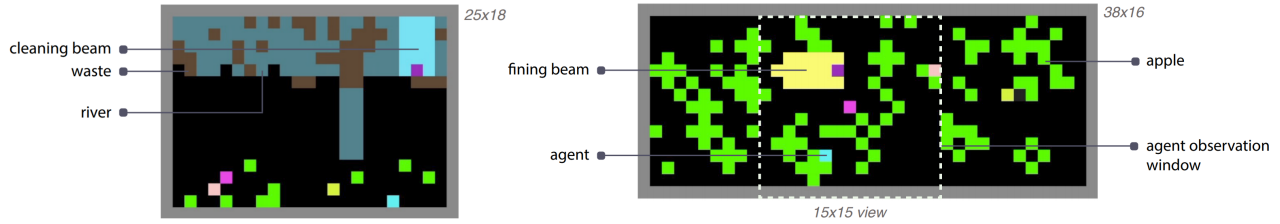


Figure 1: The two SSD environments, *Cleanup* (left) and *Harvest* (right). Agents can exploit other agents for immediate payoff, but at the expense of the long-term collective reward of the group. Reproduced with permission from **?**.

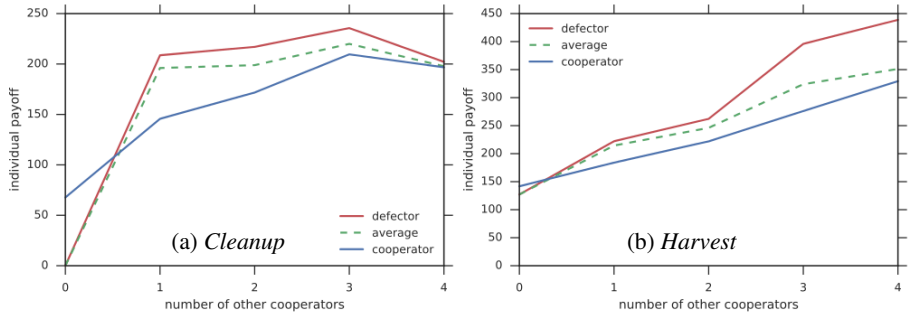


Figure 2: Schelling diagrams for the two social dilemma tasks show that an individual agent is motivated to defect, though everyone benefits when more agents cooperate. Reproduced with permission from **?**.

must efficiently time harvesting apples and cleaning the river, and allow agents cleaning the river a chance to consume apples. In *Harvest*, agents must spatially distribute their harvesting, and abstain from consuming apples too quickly in order to harvest sustainably. The code for these games, including hyperparameter settings and apple and waste respawn probabilities, can be found at https://github.com/eugenevinitsky/sequential_social_dilemma_games.

The reward structure of the games is shown in Figure 2, which gives the Schelling diagram for both SSD tasks under investigation. A Schelling diagram (**??**) depicts the relative payoffs for a single agent's strategy given a fixed number of other agents who are cooperative. These diagrams show that all agents would benefit from learning to cooperate, because even the agents that are being exploited get higher reward than in the regime where all agents defect. However, traditional RL agents struggle to learn to cooperate and solve these tasks effectively (**?**).

3. Additional experiment - Box Trapped

As a proof-of-concept experiment to test whether the influence reward works as expected, we constructed a special environment, shown in Figure 3. In this environment, one agent (teal) is trapped in a box. The other agent (purple) has a special action it can use to open the box... or it can simply choose to consume apples, which exist outside the box and are inexhaustible in this environment.

As expected, a vanilla A3C agent learns to act selfishly; the purple agent will simply consume apples, and chooses the *open box* action in 0% of trajectories once the policy has converged. A video of A3C agents trained in this environment is available at: https://youtu.be/C8SE9_YKzxI, which shows that the purple agent leaves its compatriot trapped in the box throughout the trajectory.

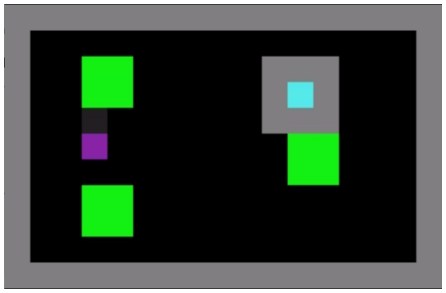


Figure 3: The *Box trapped* environment in which the teal agent is trapped, and the purple agent can release it with a special *open box* action.

In contrast, an agent trained with the social influence reward chooses the *open box* action in 88% of trajectories, releasing its fellow agent so that they are both able to consume apples. A video of this behavior is shown at: https://youtu.be/Gfo248-qt3c. Further, as Figure 4 reveals, the purple influencer agent usually chooses to open the box within the first few steps of the trajetory, giving its fellow agent more time to collect reward.

Most importantly though, Figure 5 shows the influence re-

ward over the course of a trajectory in the *Box trapped* environment. The agent chooses the *open box* action in the second timestep; at this point, we see a corresponding spike in the influence reward. This reveals that the influence reward works as expected, incentivizing an action which has a strong — and in this case, prosocial — effect on the other agent's behavior.

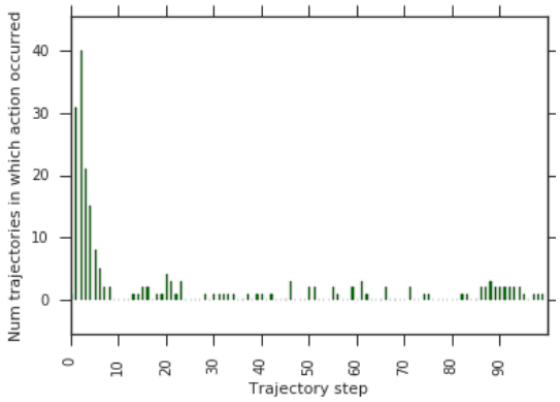


Figure 4: Number of times the *open box* action occurs at each trajectory step over 100 trajectories.

Figure 5: Influence reward over a trajectory in *Box trapped*. An agent gets high influence for letting another agent out of the box in which it is trapped.

4. Implementation details

All models are trained with a single convolutional layer with a kernel of size 3, stride of size 1, and 6 output channels. This is connected to two fully connected layers of size 32 each, and an LSTM with 128 cells. We use a discount factor $\gamma = .99$. The number of agents N is fixed to 5.

In addition to the comparison function used to compute influence (e.g. KL-divergence, PMI, JSD), there are many other hyperparameters that can be tuned for each model. We use a random search over hyperparameters, ensuring a fair comparison with the search size over the baseline parameters that are shared with the influence models. For all models we search for the optimal entropy reward and learning rate, where we anneal the learning rate from an initial value `lr_init` to `lr_final`. The below sections give the parameters found to be most effective for each of the three experiments.

4.1. Basic influence hyperparameters

In this setting we vary the number of influencers from $1-4$, the influence reward weight β, and the number of curriculum steps over which the weight of the influence reward is linearly increased C. In this setting, since we have a centralised controller, we also experiment with giving the influence reward to the agent being influenced as well, and find that this sometimes helps. This 'influencee' reward is not used in the other two experiments, since it precludes independent training. The hyperparameters found to give the best performance for each model are shown in Table 1.

4.2. Communication hyperparameters

Because the communication models have an extra A2C output head for the communication policy, we use an additional entropy regularization term just for this head, and apply a weight to the communication loss in the loss function. We also vary the number of communication symbols that the agents can emit, and the size of the linear layer that connects the LSTM to the communication policy layer, which we term the communication embedding size. Finally, in the communication regime, we experiment to setting the weight on the extrinsic reward E, α, to zero. The best hyperparameters for each of the communication models are shown in Table 2.

4.3. Model of other agents (MOA) hyperparameters

The MOA hyperparameters include whether to only train the MOA with cross-entropy loss on the actions of agents that are visible, and how much to weight the supervised loss in the overall loss of the model. The best hyperparameters are shown in Table 3.

4.4. Communication analysis

The speaker consistency metric is calculated as:

$$\sum_{k=1}^{N} 0.5[\sum_{c} 1 - \frac{H(p(a^k|m^k=c))}{H_{max}} + \sum_{a} 1 - \frac{H(p(m^k|a^k=a))}{H_{max}}], \quad (5)$$

where H is the entropy function and H_{max} is the maximum entropy based on the number of discrete symbols or actions.

Hyperparameter	Cleanup			Harvest		
	A3C baseline	Visible actions baseline	Influence	A3C baseline	Visible actions baseline	Influence
Entropy reg.	.00176	.00176	.000248	.000687	.00184	.00025
lr_init	.00126	.00126	.00107	.00136	.00215	.00107
lr_end	.000012	.000012	.000042	.000028	.000013	.000042
Number of influencers	-	3	1	-	3	3
Influence weight β	-	0	.146	-	0	.224
Curriculum C	-	-	140	-	-	140
Policy comparison	-	-	JSD	-	-	PMI
Influencee reward	-	-	1	-	-	0

Table 1: Optimal hyperparameter settings for the models in the basic influence experiment.

Hyperparameter	Cleanup			Harvest		
	A3C baseline	Comm. baseline	Influence comm.	A3C baseline	Comm. baseline	Influence comm.
Entropy reg.	.00176	.000249	.00305	.000687	.000174	.00220
lr_init	.00126	.00223	.00249	.00136	.00137	.000413
lr_end	.000012	.000022	.0000127	.000028	.0000127	.000049
Influence weight β	-	0	2.752	-	0	4.825
Extrinsic reward weight α	-	-	0	-	-	1.0
Curriculum C	-	-	1	-	-	8
Policy comparison	-	-	KL	-	-	KL
Comm. entropy reg.	-	-	.000789	-	-	.00208
Comm. loss weight	-	-	.0758	-	-	.0709
Symbol vocab size	-	-	9	-	-	7
Comm. embedding	-	-	32	-	-	16

Table 2: Optimal hyperparameter settings for the models in the communication experiment.

The goal of the metric is to measure how much of a 1:1 correspondence exists between a speaker's action and the speaker's communication message.

5. Additional results

5.1. Basic influence emergent communication

Figure 6 shows an additional moment of high influence in the *Cleanup* game. The purple influencer agent can see the area within the white box, and therefore all of the apple patch. The field-of-view of the magenta influencee is outlined with the magenta box; it cannot see if apples have appeared, even though it has been cleaning the river, which is the action required to cause apples to appear. When the purple influencer turns left and does not move towards the apple patch, this signals to the magenta agent that no apples have appeared, since otherwise the influence would move right.

5.2. Optimizing for collective reward

In this section we include the results of training explicitly prosocial agents, which directly optimize for the collective reward of all agents. Previous work (e.g. **?**) has shown that training agents to optimize for the rewards of other agents can help the group to obtain better collective outcomes.

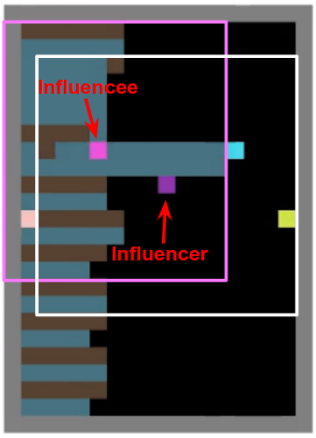


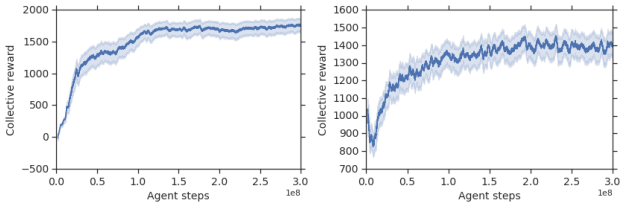
Figure 6: A moment of high influence between the purple influencer and magenta influencee.

Following a similar principle, we implemented agents that optimize for a convex combination of their own individual reward e_t^k and the collective reward of all other agents, $\sum_{i=1,i\neq k}^{N} e_t^i$. Thus, the reward function for agent k is $r_t^k = e_t^k + \eta \sum_{i=1,i\neq k}^{N} e_t^i$. We conducted the same hyperparameter search over the parameters mentioned in Section 4.1 varying the weight placed on the collective reward, $\eta \in [0, 2]$.

As expected, we find that agents trained to optimize for collective reward attain higher collective reward in both

Hyperparameter	Cleanup			Harvest		
	A3C baseline	MOA baseline	Influence MOA	A3C baseline	MOA baseline	Influence MOA
Entropy reg.	.00176	.00176	.00176	.000687	.00495	.00223
lr_init	.00126	.00123	.00123	.00136	.00206	.00120
lr_end	.000012	.000012	.000012	.000028	.000022	.000044
Influence weight β	-	0	.620	-	0	2.521
MOA loss weight	-	1.312	15.007	-	1.711	10.911
Curriculum C	-	-	40	-	-	226
Policy comparison	-	-	KL	-	-	KL
Train MOA only when visible	-	False	True	-	False	True

Table 3: Optimal hyperparameter settings for the models in the model of other agents (MOA) experiment.



(a) *Cleanup* (b) *Tragedy*

Figure 7: Total collective reward obtained by agents trained to optimize for the collective reward, for the 5 best hyperparameter settings with 5 random seeds each. Error bars show a 99.5% confidence interval (CI) computed within a sliding window of 200 agent steps.

Cleanup and *Harvest*, as is shown in Figure 7. In both games, the optimal value for $\eta = 0.85$. Interestingly, however, the equality in the individual returns for these agents is extremely low. Across the hyperparameter sweep, no solution to the *Cleanup* game which scored more than 20 points in terms of collective return was found in which all agents scored an individual return above 0. It seems that in *Cleanup*, when agents are trained to optimize for collective return, they converge on a solution in which some agents never receive any reward.

Note that training agents to optimize for collective reward requires that each agent can view the rewards obtained by other agents. As discussed previously, the social influence reward is a novel way to obtain cooperative behavior, that does not require making this assumption.

5.3. Performance comparison between models and related work

Table 4 presents the final collective reward obtained by each of the models tested in the experiments presented in the paper. We see that in several cases, the influence agents are even able to out-perform the state-of-the-art results on these tasks reported by (**?**), despite the fact that the solution proposed by (**?**) requires that agents can view other agents' rewards, whereas we do not make this assumption, and instead only require that agents can view each others' actions.

5.4. Collective reward and equality

It is important to note that collective reward is not always the perfect metric of cooperative behavior, a finding that was also discovered by **?** and emphasized by **?**. In the case, we find that there is a spurious solution to the *Harvest* game, in which one agent fails to learn and fails to collect any apples. This leads to very high collective reward, since it means there is one fewer agent that can exploit the others, and makes sustainable harvesting easier to achieve. Therefore, for the results shown in the paper, we eliminate any random seed in *Harvest* for which one of the agents has failed to learn to collect apples, as in previous work (**?**).

However, here we also present an alternative strategy for assessing the overall collective outcomes: weighting the total collective reward by an index of equality of the individual rewards. Specifically, we compute the Gini coefficient over the N agents' individual environmental rewards e_t^k:

$$G = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} |e_t^i - e_t^j|}{2N \sum_{i=1}^{N} e_t^i}, \tag{6}$$

which gives us a measure of the inequality of the returns, where $G \in [0, 1]$, with $G = 0$ indicating perfect equality. Thus, $1 - G$ is a measure of equality; we use this to weight the collective reward for each experiment, and plot the results in Figure 8. Once again, we see that the influence models give the highest final performance, even with this new metric.

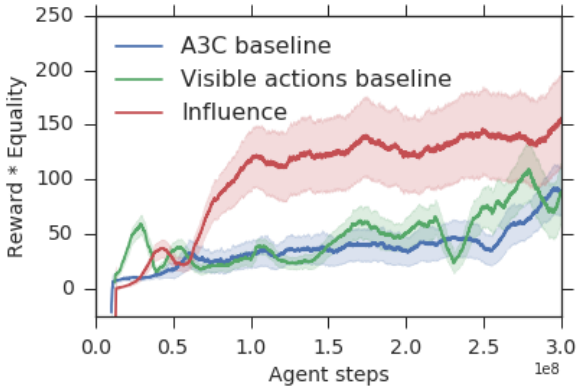
5.5. Collective reward over multiple hyperparameters

Finally, we would like to show that the influence reward is robust to the choice of hyperparameter settings. Therefore, in Figure 9, we plot the collective reward of the top 5 best hyperparameter settings for each experiment, over 5 random seeds each. Once again, the influence models result in higher

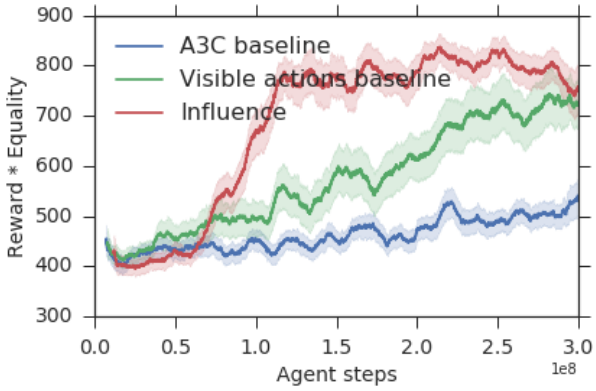
	Cleanup	Harvest
A3C baseline	89	485
Inequity aversion (**?**)	275	750
Influence - Basic	190	**1073**
Influence - Communication	166	**951**
Influence - Model of other agents	**392**	588

Table 4: Final collective reward over the last 50 agent steps for each of the models considered. Bolded entries represent experiments in which the influence models significantly outperformed the scores reported in previous work on *inequity aversion*(**?**). This is impressive, considering the *inequity averse* agents are able to view all other agents' rewards. We make no such assumption, and yet are able to achieve similar or superior performance.

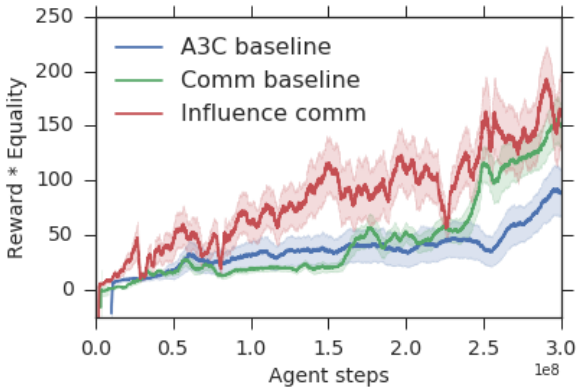
collective reward, which provides evidence that the model is robust to the choice of hyperparameters.



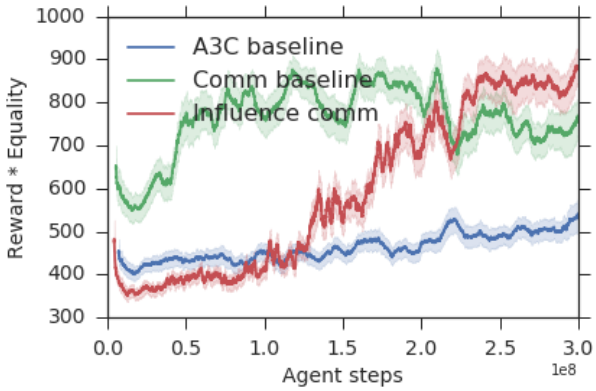
(a) *Cleanup* - Basic influence



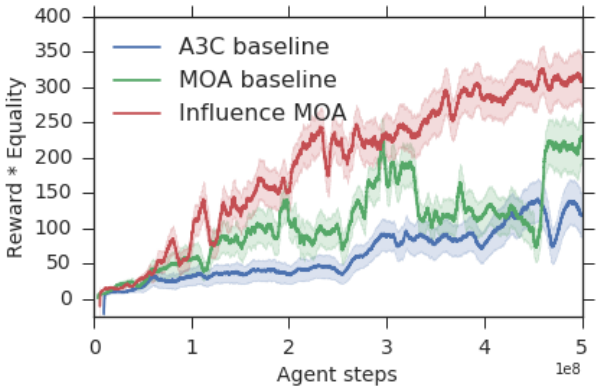
(b) *Harvest* - Basic influence



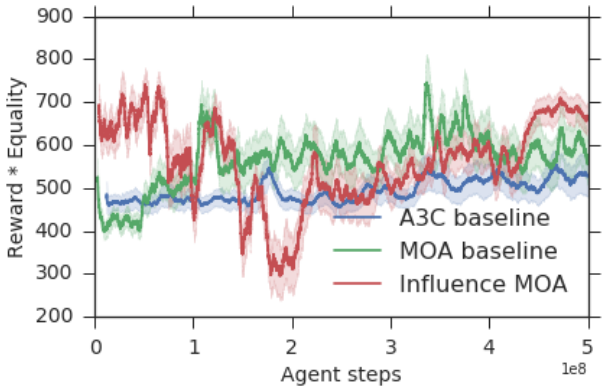
(c) *Cleanup* - Communication



(d) *Harvest* - Communication

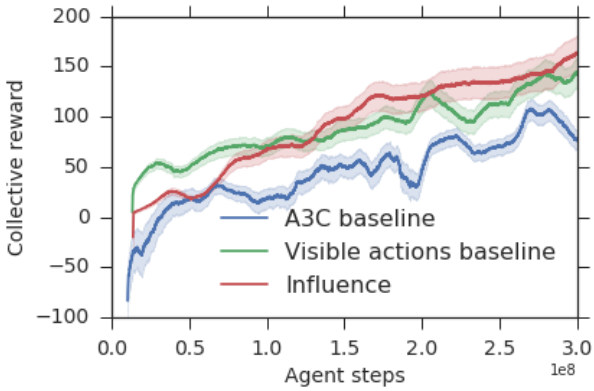


(e) *Cleanup* - Model of other agents

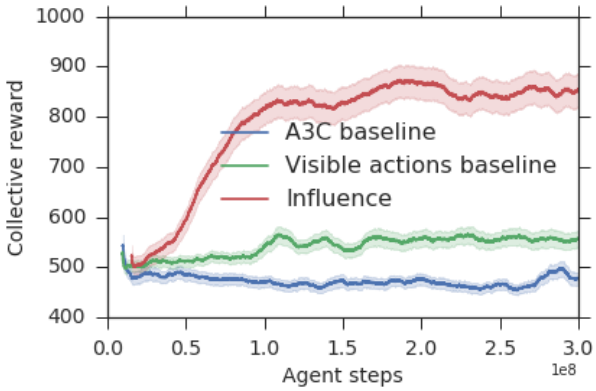


(f) *Harvest* - Model of other agents

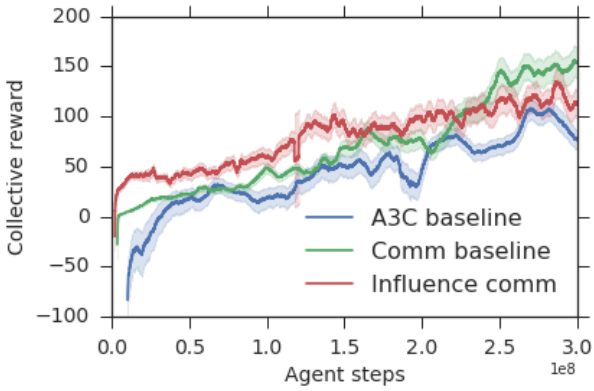
Figure 8: Total collective reward times equality, $R * (1 - G)$, obtained in all experiments. Error bars show a 99.5% confidence interval (CI) over 5 random seeds, computed within a sliding window of 200 agent steps. Once again, the models trained with influence reward (red) significantly outperform the baseline and ablated models.



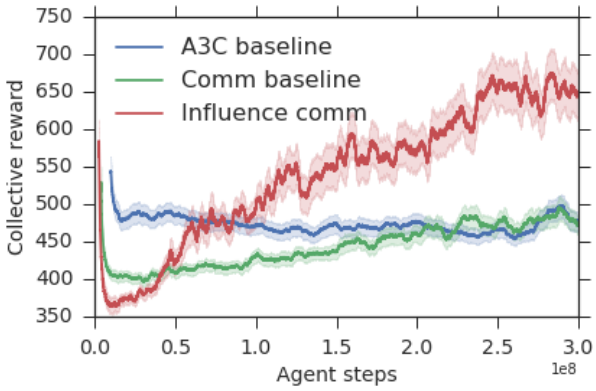
(a) *Cleanup* - Basic influence



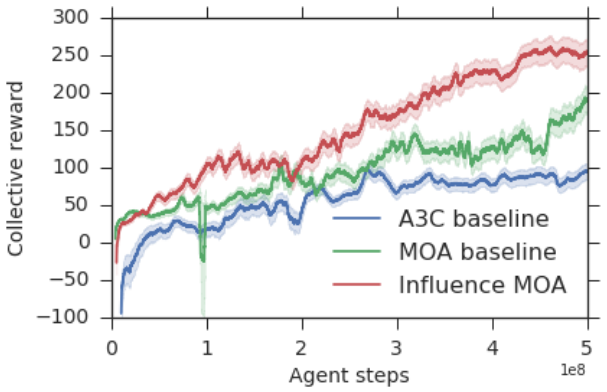
(b) *Harvest* - Basic influence



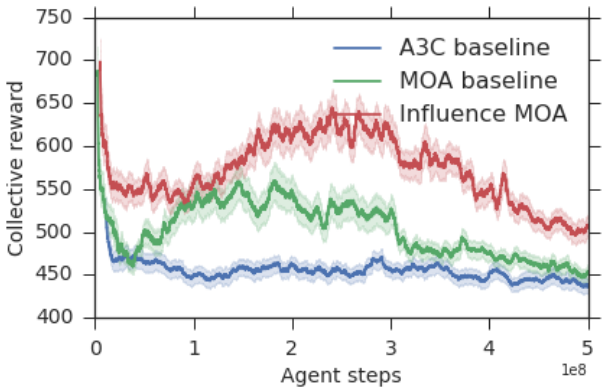
(c) *Cleanup* - Communication



(d) *Harvest* - Communication



(e) *Cleanup* - Model of other agents



(f) *Harvest* - Model of other agents

Figure 9: Total collective reward over the top 5 hyperparameter settings, with 5 random seeds each, for all experiments. Error bars show a 99.5% confidence interval (CI) computed within a sliding window of 200 agent steps. The influence models still maintain an advantage over the baselines and ablated models, suggesting the technique is robust to the hyperparameter settings.