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:

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

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

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

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. Strouse et al. (2018)),

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

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. 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. 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

[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.

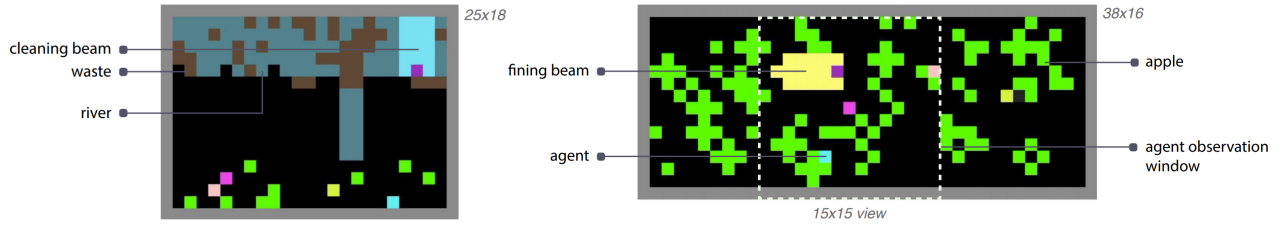


Figure 1: The two SSD environments, *Cleanup* (left) and *Harvest* (right). Agents can exploit other agents for immediate payoff, but at the expense of the long-term collective reward of the group. Reproduced with permission from Hughes et al. (2018).

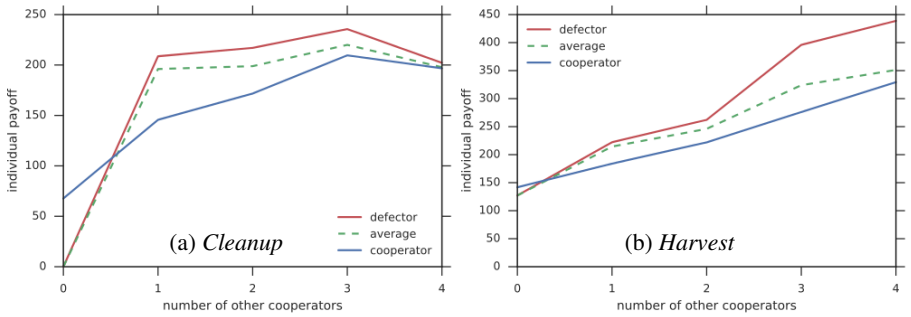


Figure 2: Schelling diagrams for the two social dilemma tasks show that an individual agent is motivated to defect, though everyone benefits when more agents cooperate. Reproduced with permission from Hughes et al. (2018).

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 2, 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).

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

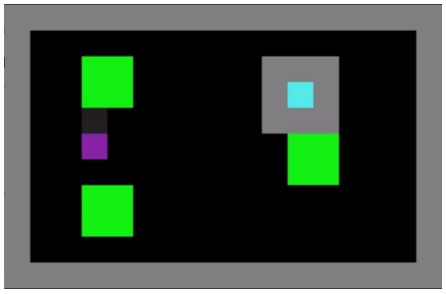


Figure 3: The *Box trapped* environment in which the teal agent is trapped, and the purple agent can release it with a special *open box* action.

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 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 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.

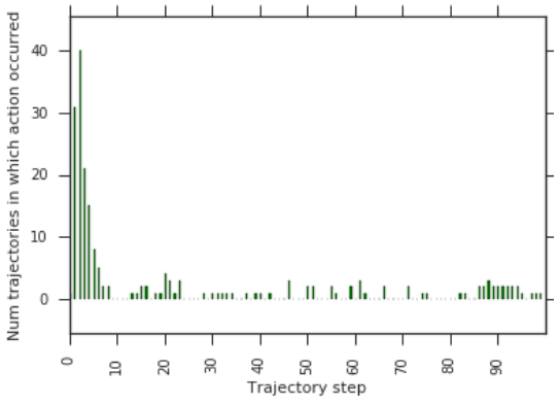


Figure 4: Number of times the *open box* action occurs at each trajectory step over 100 trajectories.

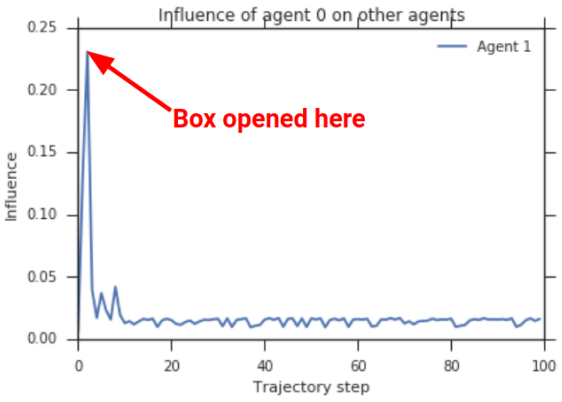


Figure 5: Influence reward over a trajectory in *Box trapped*. An agent gets high influence for letting another agent out of the box in which it is trapped.

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.

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.

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. 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.

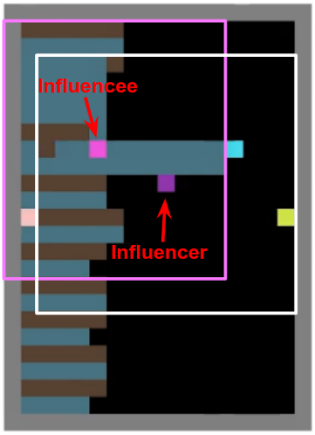


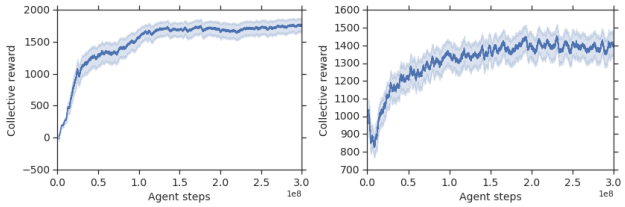
Figure 6: A moment of high influence between the purple influencer and magenta influencee.

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. Peysakhovich & Lerer (2018)) has shown that training agents to optimize for the rewards of other agents can help the group to obtain

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.

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 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 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 (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.

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 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:

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

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

	Cleanup	Harvest
A3C baseline	89	485
Inequity aversion (Hughes et al., 2018)	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.

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

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.

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.

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.

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.

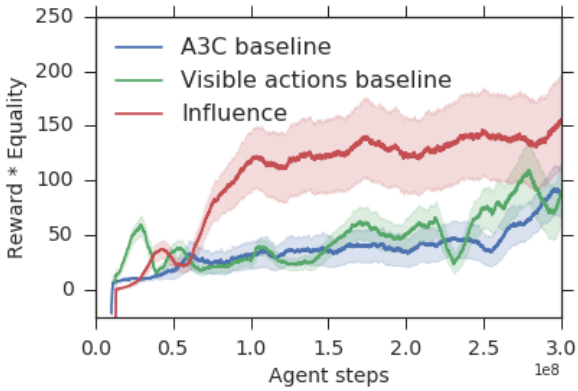
Lizier, J. T. and Prokopenko, M. Differentiating information transfer and causal effect. *The European Physical Journal B*, 73(4):605–615, 2010.

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.

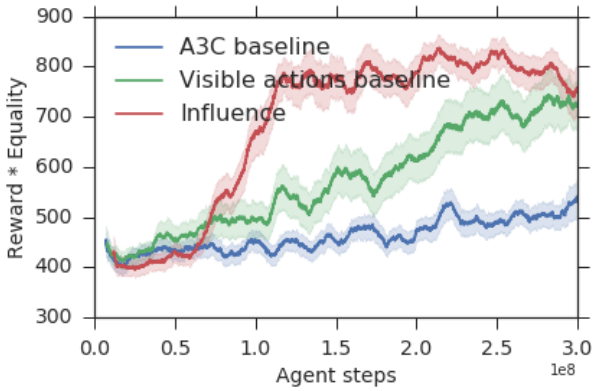
Peysakhovich, A. and Lerer, A. Prosocial learning agents solve generalized stag hunts better than selfish ones. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2043–2044. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

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.

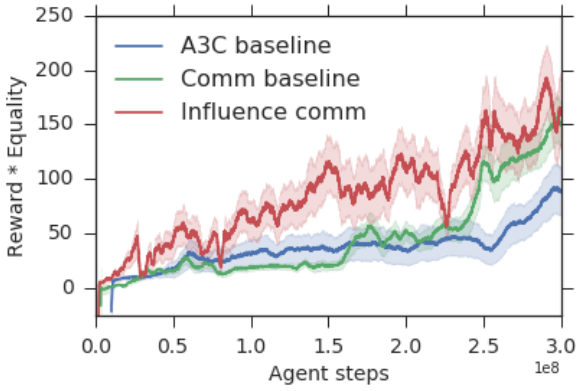
Strouse, D., Kleiman-Weiner, M., Tenenbaum, J., Botvinick, M., and Schwab, D. Learning to share and hide intentions using information regularization. *arXiv preprint arXiv:1808.02093*, 2018.



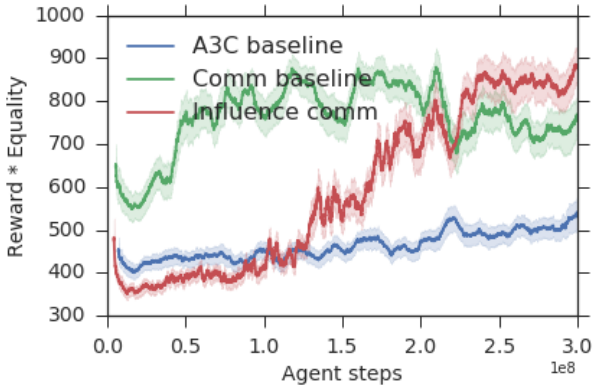
(a) *Cleanup* - Basic influence



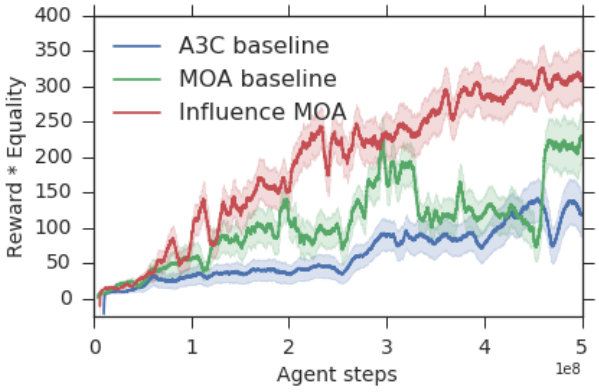
(b) *Harvest* - Basic influence



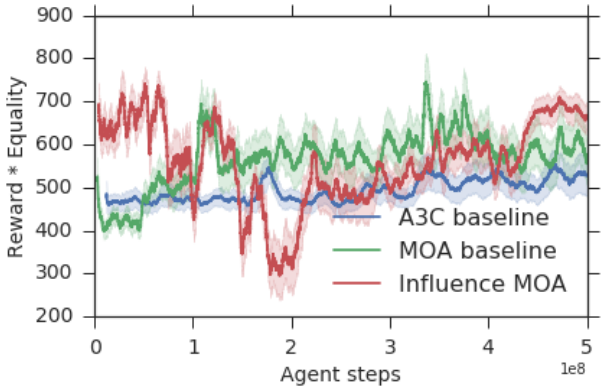
(c) *Cleanup* - Communication



(d) *Harvest* - Communication

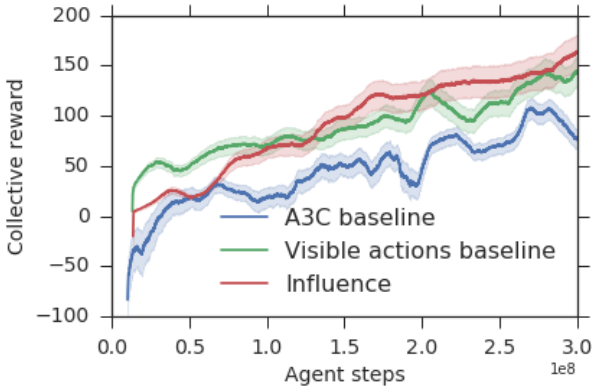


(e) *Cleanup* - Model of other agents

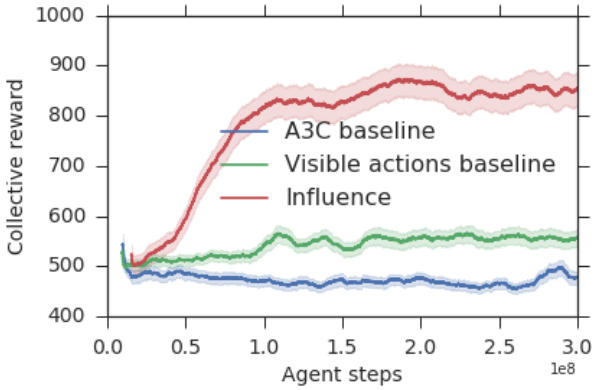


(f) *Harvest* - Model of other agents

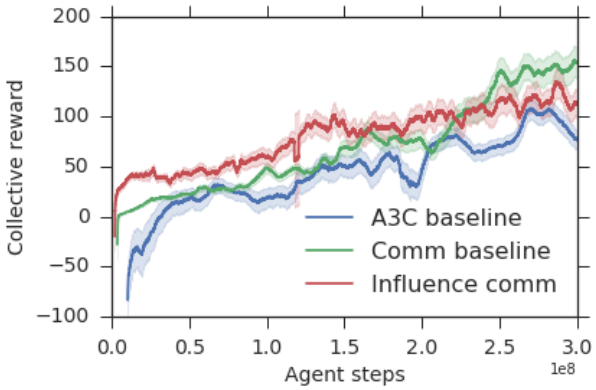
Figure 8: Total collective reward times equality, $R * (1 - G)$, obtained in all experiments. Error bars show a 99.5% confidence interval (CI) over 5 random seeds, computed within a sliding window of 200 agent steps. Once again, the models trained with influence reward (red) significantly outperform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



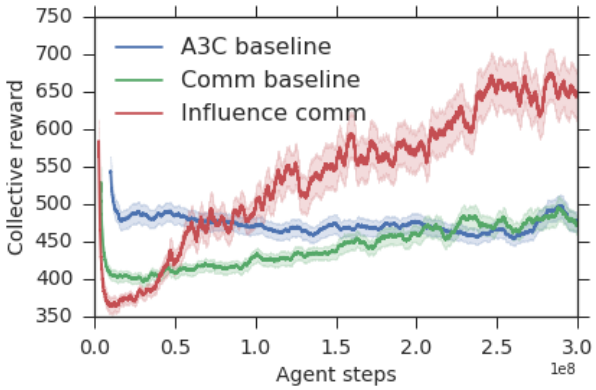
(a) *Cleanup* - Basic influence



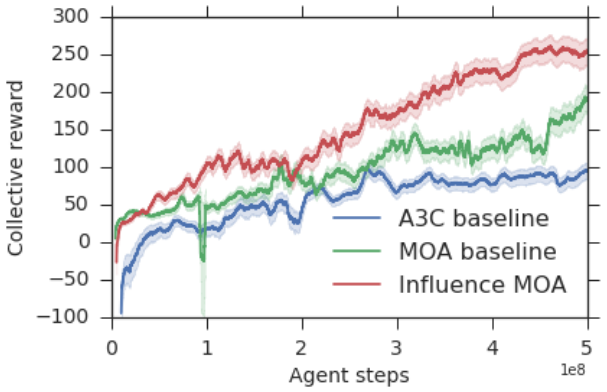
(b) *Harvest* - Basic influence



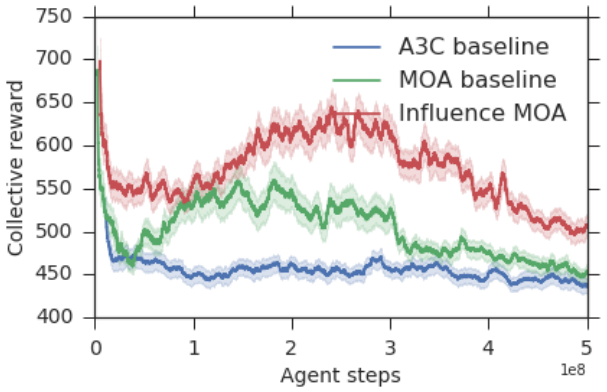
(c) *Cleanup* - Communication



(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.