Influence Based Reward Shaping Without A Heuristic

Everardo Gonzalez
Oregon State University
Corvallis, Oregon, United States
gonzaeve@oregonstate.edu

Siddarth Iyer
Oregon State University
Corvallis, Oregon, United States
viswansi@oregonstate.edu

Kagan Tumer
Oregon State University
Corvallis, Oregon, United States
kagan.tumer@oregonstate.edu

ABSTRACT

Learning based approaches have been used to coordinate multiagent systems in a variety of settings. A key challenge in multiagent learning is that agents are rewarded as a team, making it difficult to determine which agents took helpful actions. Influence based reward shaping helps address this by shaping each agent's reward signal to include credit for their actions as well as the actions of teammates they influenced. However, this requires a domain-specific definition of influence to determine who is influencing who, which may not be straightforward to define. This work-in-progress takes a step towards developing a domain-agnostic approach using causal influence. Our preliminary results show that we can tease out how one agent's actions affect its teammate's actions by simulating counterfactual scenarios.

KEYWORDS

Multiagent Systems, Reward Shaping, Social Influence

1 INTRODUCTION

Multiagent learning has been applied to many domains to achieve highly coordinated behaviors, including warehouse management [4, 12], ocean monitoring [9, 10], and space exploration [13]. A key challenge in multiagent learning is that the contributions of every agent are mixed together in a single team reward. This makes it necessary to determine who gets credit for what portion of the team reward. Otherwise, each agent tries to optimize a noisy team reward that its individual actions have little control over.

Reward shaping helps address this structural credit assignment problem by isolating an agent's impact on the team reward [2, 6, 11]. Indirect difference rewards in particular give each agent a shaped, agent-specific reward based on that agent's actions as well as the actions of teammates it influenced. The core structure of the indirect difference reward computes the difference between the team reward and a counterfactual team reward that excludes the actions of an agent and the teammates it influenced. This helps agents learn to take actions to directly improve the team reward as well as influential actions to support their teammates.

Unfortunately, this requires a domain-specific definition of influence to determine which teammates an agent is influencing. This limits the application of indirect difference rewards to problems where the type of influence necessary for high team performance is known apriori. Figure 1 shows various influential behaviors that a rover and drone team might discover. If we restrict the definition of influence to only include some of these behaviors, then we may miss helpful influential behaviors that we didn't account for.

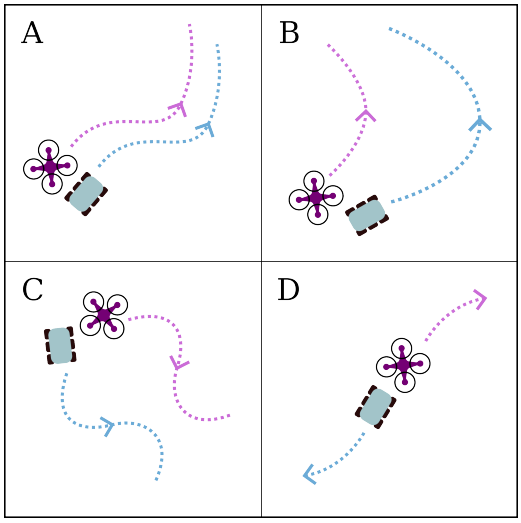

Figure 1: A rover and drone team may discover a variety of coordinated behaviors that require their actions to influence one another. These might include following (A), amplifying (B), mirroring (C), or spreading out (D). With a narrowly defined influence heuristic, agents may miss out on discovering different types of influential behaviors.

This paper is a work-in-progress that takes a step towards developing a domain-agnostic influence heuristic for credit assignment. Rather than defining a narrow heuristic and learning "influential" policies based on this metric, we can borrow ideas from social influence to measure the impact of an agent's action on its teammate's actions. The key insight of this preliminary work is that we can tease out what influence means by sampling counterfactual actions from an already coordinated team. This means moving forward we may train agents to define influence more adaptively by inferring what constitutes influence based on previous experiences.

Our contribution so far is an exploration of how we can determine if an agent is influencing its teammate using counterfactual actions. We also include immediate next steps necessary to develop and integrate a general influence measurement into reward shaping based on causal influence. From an implementation perspective, this makes influence based reward shaping heuristic free.

Preliminary results highlight how an agent's state and action distribution in a highly coordinated team can change drastically based on a teammate's influence. This suggests that causal influence can effectively measure the impact of an agent's influence without domain knowledge. By integrating causal influence into a shaped reward, we may bias agents towards exploring influential behaviors to become part of a highly coordinated team.

Proc. of the Adaptive and Learning Agents Workshop (ALA 2025), Avalos, Aydeniz, Müller, Mohammedalamen (eds.), May 19 – 20, 2025, Detroit, Michigan, USA, ala-workshop.github.io. 2025.

2 BACKGROUND

2.1 Reward Shaping

Reward shaping techniques take a team reward and modify it for individual agents [1, 5]. Structural credit assignment becomes crucial when many agents are interacting, as the team reward is the same for all agents. This is regardless of each agent's specific contribution, making it poorly suited for individual agents.

2.1.1 Difference Rewards. Difference rewards address structural credit assignment by providing an agent-specific reward that is aligned with the team reward and sensitive to an individual agent's actions [1, 3]. By comparing the actual team reward with a counterfactual team reward where that agent is removed from the team, we can calculate a difference reward that captures that individual agent's contribution to the team. A difference reward can be computed according to the following equation.

$$D_i = G(T) - G(T_{-i}) \tag{1}$$

D_i is the difference reward for agent i. T is the joint trajectory of the agents in the team. $G(T)$ is the team reward, and $G(T_{-i})$ is a counterfactual team reward where agent i's states and actions are removed from the joint trajectory. D is effective when the removal of an agent's actions directly changes the team reward. However, if removing this agent does not create an immediate difference between the team reward and the counterfactual reward, then D provides completely uninformative feedback. Influential actions from an agent can affect the actions that its teammates take, but may not directly affect the team reward. This means D is not effective for capturing and rewarding influence between agents.

2.2 Influence Based Reward Shaping

2.2.1 Indirect Difference Reward. Indirect difference rewards combine an influence heuristic with the structure from difference rewards in order to give credit for indirect impacts [7]. Consider agents that provide support to help teammates accomplish tasks, but cannot accomplish any tasks themselves. These agents would not receive any credit through D because their actions do not directly impact the team reward. Instead these agents need to be credited based on the direct impacts of the agents they influenced. An indirect difference reward provides this credit using the following equation.

$$D_i^{IND} = G(T) - G(T_{-F_i}) \tag{2}$$

D_i^{IND} is the indirect difference reward for agent i. The key distinction from the difference reward is the F_i term. F_i is a set of agents whose states and actions are removed from the joint trajectory. This is what makes it possible for agent i to get credit for the direct impacts of other agents. F_i can be computed either statically or dynamically. This means that either the same set of agents are removed at each timestep, or influence can be recomputed at each timestep so that a different set of agents are removed.

2.2.2 Influence Heuristic. In order to determine which teammates an agent is influencing, we need a domain-specific heuristic. This heuristic uses the joint-state to determine if one agent is influencing another agent. Prior work has only considered a distance threshold as an influence heuristic. If two agents are within that threshold, then they count as influencing each other. Otherwise, they would not count as influencing each other.

To determine the set of agents that are influenced by agent i, we would begin by looking at the joint state of all agents. We apply our influence heuristic to each agent in the team. If it indicates that there is influence between agent i and another agent, then we add that agent to the set F_i. If we are determining F_i dynamically, then we go through this process at each timestep in the joint trajectory. If we are instead determining F_i statically, then we aggregate influence across an entire joint trajectory, and only the agent that influenced a teammate the most includes that teammate in F_i.

2.3 Social Influence as Intrinsic Motivation

Previous work has used influence as an intrinsic motivation reward to incentivize exploration [8]. This causal influence reward is provided in addition to the environment reward at each timestep based on the amount an agent shifts another agent's action distribution. More concretely, if there are 2 agents, agent k (the influencer) and agent j (the influencee), k's causal influence can be calculated according to the following equation:

$$\begin{aligned} c_t^k &= D_{KL}\left[p(a_t^j \mid a_t^k, s_t^j) \middle\| \sum_{\tilde{a}_t^k} p(a_t^j \mid \tilde{a}_t^k, s_t^j) p(\tilde{a}_t^k \mid s_t^j)\right] \\ &= D_{KL}\left[p(a_t^j \mid a_t^k, s_t^j) \middle\| p(a_t^j \mid s_t^j)\right] \end{aligned} \tag{3}$$

In this equation, since agent j's policy is conditioned on its own state as well as agent k's action, j's marginal policy (the right side) can be derived by sampling counterfactual actions of agent k. After computing this, the impact of agent k's action on the action taken by agent j can be computed using KL Divergence. This value is then provided to agent k along with its extrinsic environment reward so the agent learns to take high-performing and influential actions that enable other agents in the system to cooperate. The probabilities necessary for these computations can be obtained centrally by using each agent's policy, or in a decentralized manner where each agent maintains a model of the other agents.

There are two distinct drawbacks to using this intrinsic motivation reward in team settings. The first is that it rewards any type of influence, regardless of whether that influence was beneficial to the team. This means an agent could be rewarded for using its influence to hinder teammates from improving the team reward. The second is that it cannot promote influence in sparse team reward settings. When only a minuscule subset of joint actions yield a non-zero reward, agents' action distributions are random. This means that an agent can be incorrectly characterized as influencing its teammates due to random action distributions, which would reward random actions rather than coordinated actions.

3 METHODS

The goal of this research is to integrate causal influence into the indirect difference reward in order to provide agent-specific rewards for various types of influential actions. By integrating causal influence into a reward shaping framework, we can automatically differentiate between helpful and spurious influence to ensure that we are rewarding helpful influence. The shaped reward is derived

from the team reward, so if an influential action does not result in a higher team reward, then it will not result in a higher shaped reward. This section details the core concepts that are a work-in-progress in this paper. We don't include experiments for all of these concepts because they are still in development.

3.1 Heuristic Free Influence

The indirect difference reward requires a domain-specific heuristic to determine which teammates an agent influenced. This heuristic limits the application and generalizability of indirect difference rewards because it requires apriori knowledge of what interactions between agents are necessary for effective coordination. We can instead use causal influence as a general heuristic that is domain-agnostic. Since this would be a general heuristic, it would no longer be necessary to define a heuristic to use indirect difference rewards. This means that from the perspective of setting up a team to learn with indirect difference rewards, there is no heuristic, so this approach effectively becomes heuristic-free.

We can use 3 as an influence heuristic. Specifically, a threshold value can be compared to the causal influence of an agent on a teammate. If the causal influence is above the threshold, then we consider that agent as influencing its teammate. To compute causal influence, we can sample counterfactual actions in our environment. This will initially require the re-simulation of agent actions, but could be modified to use models of other agents instead.

3.2 Re-Simulating Influential Behaviors

Normally in reward shaping, there is no re-simulation of joint policies. Neither the direct nor indirect difference reward require joint policies to be re-simulated. When we remove agent states and actions from a joint trajectory in reward shaping, we do not re-simulate in order to see what the other agents would have done differently. We simply recompute the team reward with a modified joint trajectory. The reason we do not re-simulate joint policies in reward shaping is because this breaks the theoretical guarantee that the shaped reward will be aligned with the team reward.

For causal influence, we need to simulate the outcomes of different counterfactual actions. This means re-simulating the environment to get either different joint actions or entirely new joint trajectories. This may seem to present a conflict with reward shaping, but if we walk through the alignment equations for the indirect difference reward, we can see why this actually does not cause any issues. The equations showing that the indirect difference reward D^{IND} is aligned with the team reward G for any agent i is shown below.

$$\begin{aligned}\frac{\partial}{\partial i} D_i^{IND}(T) &= \frac{\partial}{\partial i}[G(T) - G(T_{-F_i})] \\ &= \frac{\partial}{\partial i} G(T) - \frac{\partial}{\partial i} G(T_{-F_i}) \\ &= \frac{\partial}{\partial i} G(T) - 0 \\ &= \frac{\partial}{\partial i} G(T)\end{aligned}$$

As long as $\frac{\partial}{\partial i} G(T_{-F_i}) = 0$, then our alignment guarantee still holds. When we re-simulate, we are not actually changing T in any way. We are generating entirely separate joint trajectories or actions that we only use to determine what agents we should include in F_i other than agent i. So long as we always include agent i in F_i, then these equations will remain true, and it doesn't matter what operations we take in order to determine the other agents that should be included in F_i.

We can take re-simulation a step further and not just re-simulate a random sample of actions, but specifically bias our sampling towards actions that have scored high in their causal influence measurement in the past. One of the benefits of wrapping causal influence into the indirect difference reward is that we do not need to worry about our agents prioritizing their influence over improving the team's performance. Only influence that has actually propagated into a higher team reward will be rewarded in the shaped reward. This means we can really focus on sampling influential actions to promote higher influence scores without fear of leading our agents away from optimizing their shared team reward.

4 EXPERIMENT SETUP

4.1 Random POI Capture Problem

We run simulations in a 2D multiagent particle domain. This domain requires the coordination of a rover and a drone to capture a randomly spawning point of interest (POI). The team reward is computed based on how close the rover gets to the POI, but the rover cannot detect the POI. Instead, the drone has the sensors necessary to detect the POI, and the rover must rely on detecting the drone in its limited observation radius to guide it to the POI. We experiment with this particular domain because agents must rely on each other's influence in order to achieve a high team reward.

The rover is outfitted with low resolution sensors to detect the drone. The sensors split a rover's observation radius into 4 quadrants. The input for a single sensor is shown below, where i' iteratively represents all of the drones in a quadrant that are within the observation radius of the rover. In our experiments there is either one drone or no drones in a quadrant.

$$S_{AGENT} = \sum_{i'} \frac{1}{distance(i, i')} \tag{4}$$

The drone is outfitted with corresponding rover sensors to detect the rover as well as low resolution POI sensors. POI sensing is computed similarly to agent sensing, as shown below, where j iteratively represents all the POIs in a quadrant that are within the observation radius of a drone. In our experiments there is either one POI or no POIs in a quadrant.

$$S_{POI} = \sum_{j} \frac{1}{distance(i, j)} \tag{5}$$

The rover and drone take [dx,dy] actions in the environment to move around and interact with each other. The team reward is computed based on how close the rover gets to the POI. Note that team reward is not affected by how the drone moves; it is only directly dependent on the rover. The drone cannot capture the POI. The team reward is captured by the evaluation function in the equation below.

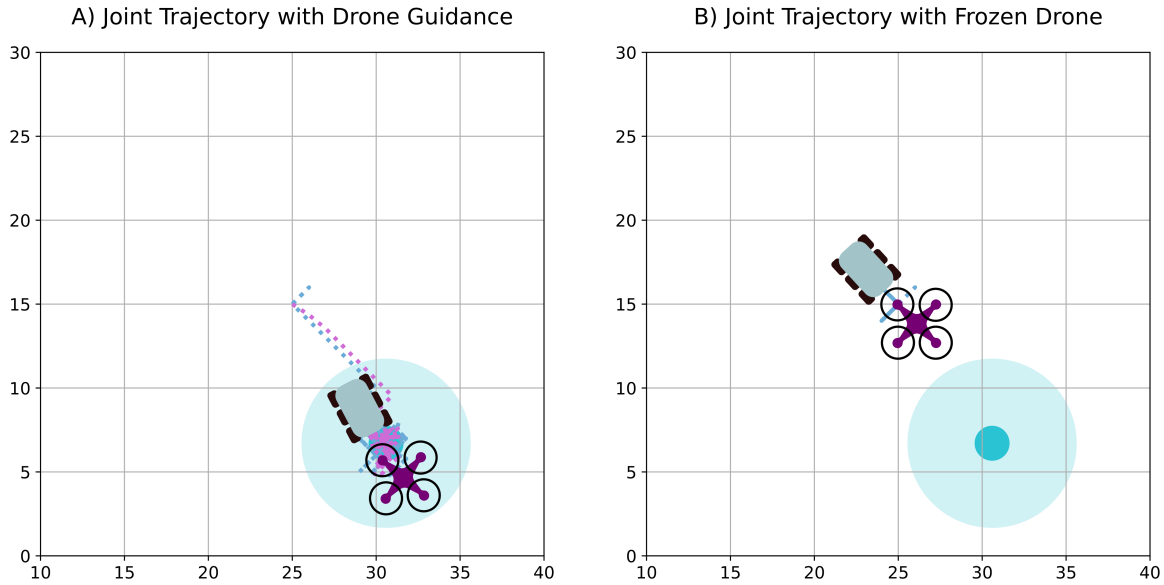

Figure 2: Two joint trajectories are shown side-by-side. The blue line represents the path taken by the rover, and the purple line represents the path taken by the drone. The blue circle represents the POI. (A) The drone and rover start at (25, 15) and the drone guides the rover to the POI. (B) The drone is frozen in place, so the rover simply stays next to the drone. From these joint trajectories, it is clear that the rover depends on the drone's influence to capture the POI, and no heuristic was necessary in order for us to draw this conclusion.

$$G(T) = \max_{t}\left(\frac{I(i,j)}{distance(i,j)}\right) \tag{6}$$

For POI j, we check how close rover i was at any point in time t. The indicator function I returns 1 if rover i was within the capture radius of POI j at time t and 0 otherwise. A high reward means that a drone guided its rover to successfully capture the POI. It is important to note that it is possible for a rover to wander away from a drone and capture a POI on its own. G does not require the drone and rover to capture the POI together. However, the rover has no sensors to detect POIs, so as soon as a rover no longer detects any of its teammates within its observation radius, it is wandering blindly in the environment.

4.2 Domain Setup

The rover's observation radius is 5, and the drone's observation radius is 100. The POI's capture radius is 5. Both agents have a maximum dx and dy of 1, and minimum dx and dy of -1.

The rover and drone begin at the (x,y) position (25,15) in a 50x50 map. The POI is randomly placed along the edge of a circle that surrounds the agents. This circle is centered at (25,15) and has a radius of 10. That means the POI is always placed 10 units away from the team, and the POI can be placed at any angle relative to the team. The team has 50 timesteps for the rover to reach the POI.

In our experiments, the team uses pre-trained neural network policies such that the drone will guide the rover to the POI. These policies are pre-trained using the indirect difference reward, and a distance based influence heuristic.

4.3 Impact of Freezing the Drone

We run 50 rollouts of these pre trained policies where the POI can spawn in random locations, and the agents have full autonomy. Full autonomy means that both agents can take actions according to their policies. We run an additional 50 rollouts with the same POI spawn locations, but this time, we freeze the drone. This means that the rover is free to take actions according to its policy, but the drone cannot take any actions. Instead, the drone remains frozen in its spawn location.

This gives us plenty of joint trajectories to help us determine how the drone influences the rover. If the rover takes the same actions regardless of whether or not the drone is frozen, then it will be clear that the rover is not influenced by the drone. However, if we see drastic changes in the rover's actions when the drone is frozen, then this gives us some insight as to what the indirect difference reward was incentivzing. This would indicate that in fact the drone learned to influence the rover and the rover learned to depend on the drone.

5 PRELIMINARY RESULTS

We break down our preliminary results into three parts.

(1) Joint Trajectories
(2) Rover State Distributions
(3) Rover Action Distributions

5.1 Joint Trajectories

The joint trajectory of the rover and drone is greatly affected by whether or not the drone is frozen. We can see in Figure 2 two joint

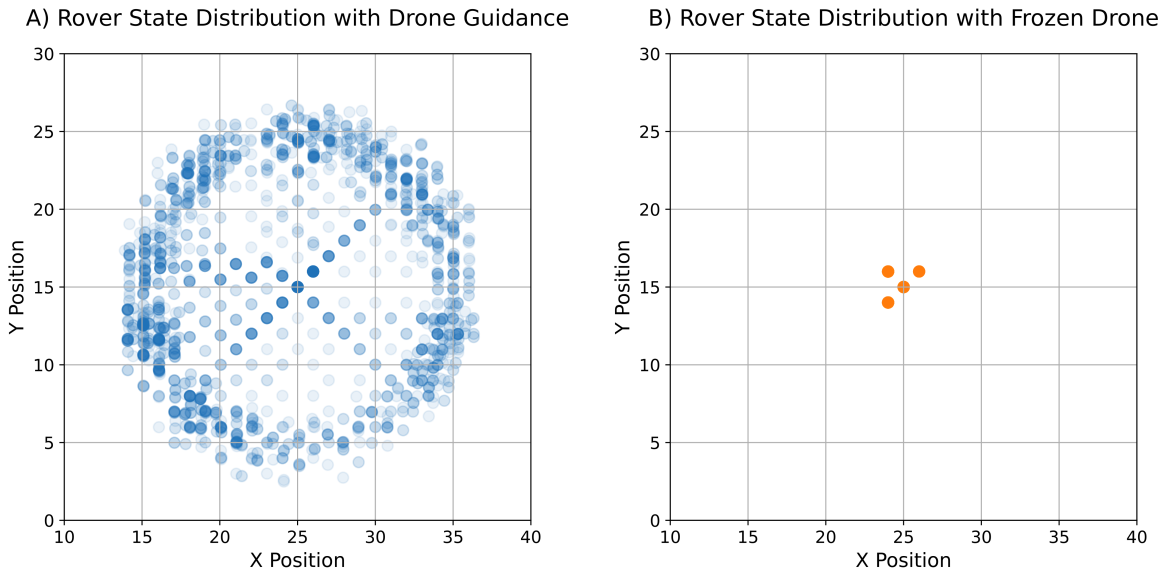

Figure 3: The rover's state distribution is shown as a scatter plot, where each point represents a position that the rover was in during all 50 rollouts. The more shaded a point is, the more frequently the rover visited that particular state. (A) When the rover can work with the drone, the rover visits many points in its state space. The rover especially visits points that along the edge of where the POI spawns, as well as points that lead to the POI. (B) When the drone is frozen, the rover only visits a few states repeatedly near its initial spawn point. We can see the rover's behavior is quantitatively different without the drone's support.

trajectories that are representative of how the team behaves when the drone can guide the rover normally (A), and when the drone is frozen in place (B). The rover clearly depends on the drone's influence in order to capture the POI, and we can see this from the change in the rover's behavior once the drone was frozen. These two joint trajectories are representative of all 50 random spawn locations for the POI. In any case where the drone can move, the drone successfully guides the rover to the POI. Conversely, when the drone is frozen, the rover stays near the starting location. We already know this team has a high performing joint policy. The fact that we can run these counterfactual simulations and get very different team behaviors suggests that causal influence may tease out a more complete picture of how these agents influence each other.

5.2 State Distributions

We can see the effect of the drone on the rover's state distribution in Figure 3. When the drone is able to move according to its policy, the rover can follow the drone throughout the map. Specifically, we can see there is a concentration of points around the edge of where the POI spawns. That means that the rover spends a lot of time in places the POI spawns when the drone can guide it. When we freeze the drone, we see a massive reduction in the spread of states visited by the rover. The rover only visits four states when the drone is frozen, and they are all around the spawn point. This is because the rover is still trying to follow the drone, even though the drone isn't moving. The result is that the rover just bounces between a few states. The stark contrast in the rover's state distribution when the drone is frozen suggests that we can quantitatively measure the influence an agent has on its teammate without resorting to a domain-specific heuristic. We don't need to know anything domain-specific for us to see the impact of the drone's actions on the rover's state distribution.

5.3 Action Distributions

Casual influence tends to focus on actions, not states, so in Figure 4, we see the effect of freezing the drone on the rover's action distribution. The rover takes many different actions when the drone is able to guide it around, but takes the same four actions repeatedly when the drone is frozen. This is the type of impact that casual influence should help us measure. In the case of the frozen drone, we are only sampling one counterfactual action for the drone. That action is setting (dx,dy) to (0,0). Casual influence should give us a more complete picture of the relationship between the rover and the drone because it uses a greater variety of counterfactual actions as samples.

6 DISCUSSION

This work-in-progress suggests that the influence necessary for a highly coordinated team may be measurable without any domain knowledge. Preliminary results show that an agent's state and action distribution can change drastically in response to counterfactual actions from a teammate. The immediate next step is to use causal influence as a measurement for influence during training. In order to replace the influence heuristic, this measurement must output 1 if an agent is influencing a teammate and 0 if not.

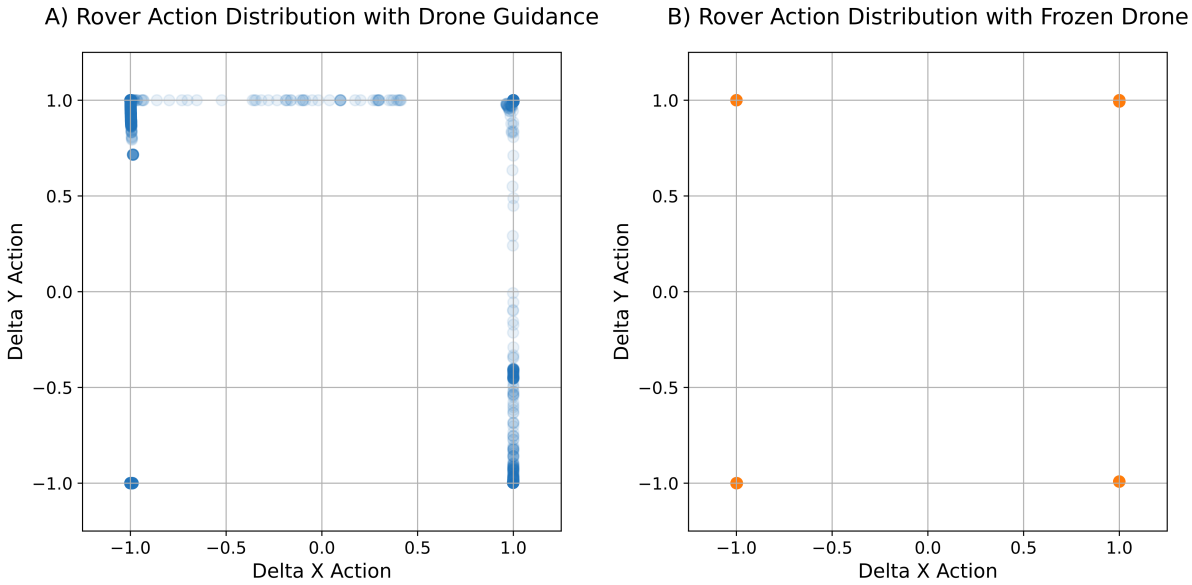

Figure 4: The rover's action distribution is shown as a scatter plot, where each point represents an action the rover took during all 50 rollouts. The more shaded a point is, the more frequently the rover took that action. (A) When the rover can work with the drone, the rover takes a variety of actions, and there is even a spectrum of actions the rover takes ranging from its maximum delta x and y values to more fine tuned movements. (B) When the drone is frozen, the rover takes the same four actions repeatedly, and there is no precision in the rover's movements. The rover is taking the maximum steps it can at each timestep. This is the type of discrepancy that casual influence could help us quantify for a shaped reward.

Since causal influence is not a binary measurement, we will apply a threshold to turn its output into a 0 or 1. We can use this measurement to determine which teammates an agent influenced and compute an influence based shaped reward that is heuristic-free.

This work additionally opens the door to broader research directions in terms of how we should be measuring influence for shaped rewards. Causal influence is just the first step. Agents might instead only use causal influence at the beginning of training, and then learn a narrower definition throughout training as they figure out their particular role on the team. Similarly, agents might combine an influence-based exploratory reward with their shaped reward in order to better balance exploring influential behaviors with their teammates and exploiting high performing behaviors. These research directions would make agents learn more adaptive definitions of influence to improve their coordination across a variety of domains, each requiring unique interactions to achieve high team performance.

REFERENCES

[1] Adrian Agogino and Kagan Tumer. 2004. Efficient Evaluation Functions for Multi-rover Systems. In *Genetic and Evolutionary Computation – GECCO 2004*, Kalyanmoy Deb (Ed.). Springer, Berlin, Heidelberg, 1–11.

[2] Jacopo Castellini, Sam Devlin, Frans A. Oliehoek, and Rahul Savani. 2022. Difference rewards policy gradients. *Neural Computing and Applications* (Nov. 2022). https://doi.org/10.1007/s00521-022-07960-5

[3] Jacopo Castellini, Sam Devlin, Frans A Oliehoek, and Rahul Savani. 2022. Difference rewards policy gradients. *Neural Computing and Applications* (2022), 1–24.

[4] Ho-Bin Choi, Ju-Bong Kim, Youn-Hee Han, Se-Won Oh, and Kwihoon Kim. 2022. MARL-Based Cooperative Multi-AGV Control in Warehouse Systems. *IEEE Access* 10 (2022), 100478–100488. https://doi.org/10.1109/ACCESS.2022.3206537

[5] Joshua Cook and Kagan Tumer. 2022. Fitness Shaping for Multiple Teams. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Boston, Massachusetts) *(GECCO '22)*. Association for Computing Machinery, New York, NY, USA, 332–340. https://doi.org/10.1145/3512290.3528829

[6] Everardo Gonzalez, Siddarth Viswanathan, and Kagan Tumer. 2024. Indirect Credit Assignment in a Multiagent System. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems* (Auckland, New Zealand) *(AAMAS '24)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2288–2290.

[7] Everardo Gonzalez, Siddarth Viswanathan, and Kagan Tumer. 2024. Influence Based Fitness Shaping for Coevolutionary Agents. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Melbourne, VIC, Australia) *(GECCO '24)*. Association for Computing Machinery, New York, NY, USA, 322–330. https://doi.org/10.1145/3638529.3654175

[8] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. 2019. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*. PMLR, 3040–3049.

[9] Maryam Kouzehgar, Malika Meghjani, and Roland Bouffanais. 2020. Multi-Agent Reinforcement Learning for Dynamic Ocean Monitoring by a Swarm of Buoys. In *Global Oceans 2020: Singapore – U.S. Gulf Coast*. 1–8. https://doi.org/10.1109/IEEECONF38699.2020.9389128

[10] Dohyun Kwon, Joohyung Jeon, Soohyun Park, Joongheon Kim, and Sungrae Cho. 2020. Multiagent DDPG-Based Deep Learning for Smart Ocean Federated Learning IoT Networks. *IEEE Internet of Things Journal* 7, 10 (2020), 9895–9903. https://doi.org/10.1109/JIOT.2020.2988033

[11] Aida Rahmattalabi, Jen Jen Chung, Mitchell Colby, and Kagan Tumer. 2016. D++: Structural credit assignment in tightly coupled multiagent domains. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 4424–4429. https://doi.org/10.1109/IROS.2016.7759651

[12] Oren Salzman and Roni Stern. 2020. Research challenges and opportunities in multi-agent path finding and multi-agent pickup and delivery problems. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. 1711–1715.

[13] Logan Yliniemi, Adrian K Agogino, and Kagan Tumer. 2014. Multirobot coordination for space exploration. *AI Magazine* 35, 4 (2014), 61–74.