# Reward Redistribution Mechanisms in Multi-agent Reinforcement Learning

Ibrahim, Aly*
McGill Univeristy
aly.ibrahim@mail.mcgill.ca

Jitani, Anirudha*
McGill University and MILA
anirudha.jitani@mail.mcgill.ca

Piracha, Daoud*
McGill University and MILA
daoud.piracha@mail.mcgill.ca

Precup, Doina
McGill University, MILA and DeepMind
dprecup@mcgill.ca

## ABSTRACT

In typical Multi-Agent Reinforcement Learning (MARL) settings, each agent acts to maximize its individual reward objective. However, for collective social welfare maximization, some agents may need to act non-selfishly. We propose a reward shaping mechanism using extrinsic motivation for achieving modularity and increased cooperation among agents in Sequential Social Dilemma (SSD) problems. Our mechanism, inspired by capitalism, provides extrinsic motivation to agents by redistributing a portion of collected rewards based on each agent's individual contribution towards team rewards. We demonstrate empirically that this mechanism leads to higher collective welfare relative to existing baselines. Furthermore, this reduces free rider issues and leads to more diverse policies. We evaluate our proposed mechanism for already specialised agents that are pre-trained for specific roles. We show that our mechanism, in the most challenging CleanUp environment, significantly outperforms two baselines (based roughly on socialism and anarchy) and accumulates 2-3 times higher rewards in an easier setting of the environment.

## KEYWORDS

AAMAS; ACM proceedings; Multi-Agent Reinforcement Learning; Extrinsic Motivation; Sequential Social Dilemmas; Cooperative MARL; Markov Games

## 1 INTRODUCTION

A key reason why society has flourished is the ability of its constituents to coordinate and cooperate. In our daily lives, we often encounter tasks which are too complex for an individual or machine to solve alone. Coordination among groups is often needed to surmount hurdles, parallelize repetitiveness, and blend complementary strengths to collectively accomplish a wider range of tasks. This problem of establishing coordination in the context of autonomous agents has remained a challenging feat. If an agent greedily chooses to optimize their own gain, we may reach sub-optimal outcomes for the community as a whole [12]. A subset of problems requiring agents that are not fully selfish is Sequential Social Dilemmas (SSDs), where there is a conflict between individual and collective interest.

We believe that a set of rules or a social structure built on top of the environment is key to achieving coordination in such problems, and hence better performance on the task at hand. Human models of governance can provide inspiration in terms of structures that can be used to ensure appropriate multi-agent coordination. In this work, we look at the environment and all the agents in analogy to a country and its citizens. The 'government' wants to maximize the GDP (Gross Domestic Product) or sum of rewards of all agents, as opposed to individual gain/rewards. However, to incentivize agents to work together is a complicated task. There are many questions that arise while trying to develop such a framework. Is there a need for a leader or influencer that masses follow to achieve a common goal? Can such a leader naturally evolve or must we introduce some form of extrinsic reward or artificial competition? Is there a need for trust (or maintaining a model of other agents)? Can agents realize if there is an adversary in the team and learn to ignore them or attenuate their negative effect? Finally, how do we go about solving coordination in a setting in which different agents might want to optimize policies which are driven by different incentives?

In the quest to answer the aforementioned questions, we propose a unified method for achieving coordination and cooperation among the agents by incentivizing the agents to behave in a selfless manner using extrinsic rewards enforced by an imaginary form of government or social structure. Extrinsic motivation [2] is a means of influencing actions by some specific rewarding outcome. Our approach borrows key concepts from the capitalistic economic model, such as taxation, and employment, to provide a way of delivering such rewards.

To demonstrate our approach, we use the Cleanup environment [6]—a public goods dilemma in which agents are rewarded for consuming apples, but they must clean a river in order for apples to grow, an activity which yields no reward. To facilitate cleaning of the river, a portion of the rewards collected by the agents are taxed and redistributed among cleaners proportional to the work they do. We call this the "capitalistic" approach. In our experiments, we also consider a variant of this approach in which agents are pre-trained to perform specialized roles (denoted by cleaning and harvesting), and we observe that the agents learn to coordinate better and faster after this pre-training phase, which leads to better social welfare compared to other approaches.
The main contributions of the paper are:

- We formulate a reward-shaping mechanism using extrinsic motivation, for achieving modularity and increased cooperation among agents in Sequential Social Dilemma (SSD) problems, without any communication overhead.

*Authors Contributed Equally to this work

- We evaluate this approach by pre-training agents to perform specific roles, which increases the speed of learning drastically and also improves the coordination amongst them.

## 2 RELATED WORK

The problem of achieving coordination in the setting of Multi-Agent Reinforcement learning [19] has been explored in various applications, including autonomous vehicles [22], traffic control [4, 23], distributed network systems [10, 13], multi-robot control [7, 16], multi-player games [20], and more. The cooperative MARL problem can be attacked using a centralized approach, thus reducing the problem to single-agent reinforcement learning over the observations and actions of all agents. Such approaches assume that a central controller has access to all the required information about all the agents instantaneously (which is not practical) and suffers from a combinatorial explosion as the number of agents grows. Some recent works [5, 14] use centralized training and decentralized execution approaches, allowing the policies to use extra information at training time, in order to simplify the learning problem. In the decentralized approaches, the agents only have a partial view of the world and the environment becomes non-stationary from an agent's perspective. Therefore, the agents must discover some form of communication protocol or signalling mechanism that enables them to coordinate their behaviour and solve the task. Moreover, it is possible to maintain an approximation of the other agents' policies and train a collection of different sub-policies to stabilize learning as shown in [14]. In [5] error derivatives between agents are back-propagated through the communication channels. In [24] a decentralized actor-critic method is used, where the actor step is performed independently, whereas, for the critic step, they propose a consensus update via communication over the network. Learning meaningful emergent communication protocols is very difficult and challenging due to limited communication channels, inaccurate credit assignment, partial observability, unreliable estimates of other agents, as was empirically shown by [1, 5, 11]. In our approach, no agent needs to have information/estimates of the observations, policies, or actions of other agents.

Recently, there have been few works that employ reinforcement learning to maximize social welfare in SSDs—[12] shows how conflict can emerge from competition over shared resources and shed light on how the sequential nature of real world social dilemmas affects cooperation. The authors in [6] show that the collective reward obtained by a group of agents in SSDs gives a clear signal about how well the agents learned to cooperate. They introduce an inequity aversion motivation, which penalizes agents if their rewards differ too much from those of the group. In [3] agents are categorized as imitators and innovators. Innovators learn purely from the environment reward. Imitators learn to match the social-mindedness level of innovators, demonstrating reciprocity. They employ a niceness network using the advantage function to calculate the niceness of an agent's action(s) or trajectory. The authors in [8] empirically show that coordination can be achieved by rewarding agents for having causal influence over other agents' actions and prove that it is equivalent to rewarding agents for having high mutual information between their actions. We use a slightly different approach, relying on extrinsic motivation to influence the behavior of agents. This is computationally more efficient than the approaches discussed above, as the actions are based only on the state observed by the agent and do not depend on counterfactual reasoning or mutual information, which require conditioning on the actions of other agents too.

## 3 BACKGROUND

### 3.1 Sequential Social Dilemmas (SSD)

A social dilemma is a situation in which individual selfishness yields a profit, until everyone chooses a selfish strategy, in which case all parties incur a loss. This imposes tensions between collective and individual rationality [21]. Each agent ideally wants to maximize their individual reward objective, but for collective social welfare maximization, some agents may need to act non-greedily with respect to their reward objectives. An individual agent can obtain higher reward in the short-term by engaging in non-cooperative behavior (and thus is greedily motivated to defect); however this will cause other agents to defect as well, leading to lower overall rewards. In many practical MARL applications, optimizing for the team's mission is more important than individual reward, hence defection can be quite damaging to overall reward.

Social dilemmas can be well understood from the theory of repeated general-sum matrix games. Specifically, consider a matrix game with the following properties of its four payoffs (Reward, Punishment, Sucker, and Temptation,) (formulated by [15]):

(1) $R > P$ : Mutual cooperation is preferred to mutual defection
(2) $R > S$ : Mutual cooperation is preferred to unilateral cooperation
(3) $2R > T + S$ : Mutual cooperation is preferred to an equal probability of unilateral cooperation and defection
(4) $T > R$: [Greed] Unilateral defection is preferred to mutual cooperation
(5) $P > S$ : [Fear] Mutual defection is preferred to unilateral cooperation

where $R$ is the reward for cooperation, $P$ is the punishment for defection, $S$ is the sucker outcome for a player who cooperates with a defector, and $T$ is the temptation outcome achieved by defecting against a cooperator.

### 3.2 Multi-Agent Reinforcement Learning for SSDs

A MARL Markov game is defined by the tuple $\langle S, T, A, r \rangle$, where each agent is trained independently to maximize its individual reward. The state of the environment at timestep $t$ is defined by $s_t \in S$. Furthermore, each agent $k$ selects an action $a_t^k \in A$ at timestep $t$. The joint action of all $N$ agents $a_t = [a_t^0, a_t^1 .... a_t^N]$ produces a transition $T(s_{t+1}|a_t, s_t)$, according to the state transition distribution. Each agent receives a reward $r_k(a_t, s_t)$, which may be dependent on actions of other agents. The trajectory over time is denoted $\tau = \{s_t, a_t, r_t\}_{t=0}^{T}$.

We consider a partially observable setting in which the $k^{th}$ 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 $\gamma$ is the discount factor.

### 3.3 Asynchronous Advantage Actor-Critic (A3C) Algorithm

In our work, a distributed asynchronous advantage actor-critic (A3C) approach will be used to train each agent $j$'s policy $\pi_j$. A3C [17] was formulated to maintain a policy $\pi(a_t|s_t;\theta)$ and an estimate of the value function $V(s_t;\phi)$. The algorithm operates in the forward view and uses the n-step return to update both the policy and the value function. The update performed by the algorithm after every $t_{\max}$ steps is seen as $\nabla_{\theta'} \log \pi(a_t|s_t;\theta') \times A(s_t, a_t;\theta,\phi)$ where $A(s_t, a_t;\theta,\phi)$ is the estimate of the advantage function defined by $\sum_{i=0}^{k-1} \gamma^i \times r_{t+1} + \gamma^k \times V(s_{t+k};\phi) - V(s_t;\phi)$, where $k$ is upper bounded by $t_{\max}$. The algorithm comprises parallel actor-learners and accumulated updates for improving training stability (see Figure 2). Even though the parameters $\theta$ of the policy and $\phi$ of the value function are shown as being separate for generality, in practice the policy and value function usually share a common layer of features. We use the same Actor and Critic Networks as [8] consisting of a convolutional layer, some fully connected layers, a Long Short Term Memory (LSTM) recurrent layer and some linear layers. All networks take images as input and output both the policy $\pi_j$ and the value function $V^{\pi_j}(s)$.

## 4 ENVIRONMENT

In the Cleanup environment [6] (Figure 1) the goal of the agents is to collect apples. Each apple collected provides a reward of **+1** to the agent which collects it. Apples spawn at a rate proportional to the cleanliness of the river. Over time, this river fills with waste, lowering the rate of apple spawning linearly. The agents can take actions to clean the waste, which provides no reward but is required to generate any apples. Therefore, the agents must be able to coordinate cleaning the river and collecting apples in order to maximize their social welfare. The episode ends after 1000 steps, and the map is reset to a random initial state. If some agents are contributing to the social good by clearing waste from the river, there is an incentive to stay in the region where the apples grow to collect apples as they spawn. However, if all players adopt this strategy, then no apples spawn and there is no reward for any agent.

## 5 APPROACH

In environments exhibiting SSDs, allowing agents to learn their policies individually can severely reduce social welfare. To overcome this we adopt ideas from economic theory as reward shaping mechanisms enforced by *a facade representing a sort of government*. Specifically, we will define two such mechanisms, which we name "Capitalism" and "Socialism", based on the economic theories which inspired them, as well as an "Anarchy" baseline, in which no "government" is present. The challenge is how to represent these economic paradigms in an SSD environment to maximize the overall reward of the agents. We are interested in incorporating ideas of taxation, wealth redistribution, elections, specialization, trust, and minimum wage to MARL, but we will focus on a subset of these issues in this work.

In particular, our model of governance must help with the following issues:

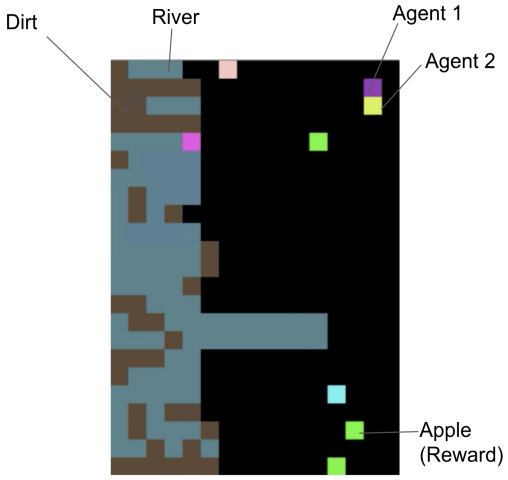



**Figure 1: Cleanup Environment [6]**

- **(Mitigate the Risk of Inflation)** Do not introduce rewards beyond what the environment offers, instead redistribute the rewards that agents receive at every time step among all agents, in a way that maximizes social welfare.
- **(Eliminate Unemployment)** Each agent needs to do meaningful work to receive a reward. Free riding should not merit earnings.
- **(Proper Credit Assignment)** Agents are rewarded proportionally to the opportunities and wealth they create or amass.

One of the main benefits of our framework, is its client-server architecture, where each agent is agnostic to the other agents (in the sense of not needing to maintain a belief over other agents' policies). The agents only communicate with a central government, to which they provide their actions and environment-rewards at each time-step. The government then gives agents the appropriate rewards to update their policies as shown in Figure 2. This allows easy utilization of the paradigms we are about to introduce in any environment.

For the remainder of this section, we introduce our main contribution developing the *Capitalism* paradigm, and a variant of it with *Specialization*, on the CleanUp environment with vanilla A3C as the underlying learning algorithm. Afterwards we introduce the *Socialism* paradigm. Next, we show how vanilla A3C can be thought of as an *Anarchy* setting. We finally discuss briefly how these ideas can be extended to other environments.

### 5.1 Capitalism

Capitalism is a form of government centralized around corporate/private ownership as a means of creating wealth. Ownership here means that agents control their labor and land, deriving their income from this ownership. This gives them the ability to operate efficiently to maximize profit. Free markets arises naturally, with a notion of supply and demand. To make this concrete, consider the CleanUp environment; if apples are abundant, agents cleaning the environment should get paid less than the harvesters. On the

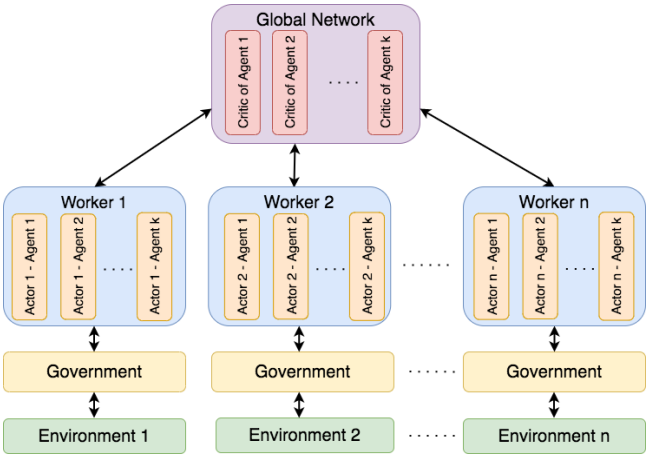



**Figure 2: At every timestep, the agent takes an action using its policy, which is dependent on its own state. It sends the action to the government, which acts as an interface between the agent and the environment. The government collects the actions of all agents, gets the rewards from the environment for all agents and makes the necessary redistribution.**

other hand, if everybody is harvesting, then apples will be in short supply, and cleaners should be paid more since they are creating more wealth. This led us to consider a notion of educating some agents to specialize as Harvester (with a minor in cleaning), and the rest as Cleaners (with a minor in harvesting) to maintain an equilibrium of supply and demand, with all agents contributing value.

We formulate our paradigm around key features including:

- Introducing the concept of *Taxation* of wealth-creating-agents (aka. **Harvesters.**)
- Introducing the concept of *Wage* given to unrewarded laborers (aka. **Cleaners**), as a pseudo-reward for their indirect contribution in increasing aggregate reward.
- Introducing the concept of Specialization of agents.

We use wage (money) as an extrinsic motivation for cleaners to perform the cleaning task. The harvesters are taxed a certain portion of their rewards and this taxed reward is redistributed among agents in proportion to the amount of cleaning work done by them in a discounted window.

#### 5.1.1 *Training/Coordination formulation for Capitalism.*

We define $C_{a,t}$ as the number of dirt cells agent $a$ cleaned in time $t$, and $H_{a,t}$ to be an indicator of whether agent $a$ collected an apple (+1) at time t or not (+0). Let $w$ be the window in the past during which a reward for cleaning is considered. Let $\gamma$ be a discounting factor for the past work of each agent and $\alpha$ be the redistribution ratio, where $\gamma, \alpha \in [0, 1]$. $W_{a,t}$ is the amount of cleaning (not rewarded) work the agent performed in the past $w$ time steps. We denote by $SW_t$ the social welfare at time $t$. We define the reward given to agent $a$ at time $t$ by $r_{a,t}$.

Then the following equations follow naturally:

$$C_{a,t} = \#(\text{CLEANED})$$

$$H_{a,t} = \mathbf{1}(\text{APPLE})$$

$$W_{a,t} = \sum_{\tau=0}^{w} \gamma^{\tau} C_{a,t-\tau}$$

$$SW_t = \sum_{a \in \text{ Agents}} H_{a,t}$$

$$r_{a,t} = \alpha H_{a,t} + (1-\alpha) SW_t * \frac{W_{a,t}}{\sum_{a \in \text{ Agents}} W_{a,t}}$$

#### 5.1.2 *Pre-training for Specialized Agents.*

We added a pre-training phase, in which we change the environmental reward structure for agents to *learn* specialized policies. Agents pre-train together in the environment, and transfer their policies during testing. For pre-training, we split agents into our chosen specializations. Our reward structure gives a reward $r = \beta$ when the agent performs their specialization, and a reward of $r = \epsilon = 1 - \beta$ for their "minor", where $\epsilon < \beta$.

Concretely, the harvester receives $\beta$ rewards for collecting apples and $\epsilon$ for cleaning a block of dirt in the river, and vice versa for the cleaners.

### 5.2 Socialism

Socialism is a form of government centralized around social ownership as a means of creating wealth. It is characterised by social ownership of the means of production and workers' self-management of enterprise. To make this concrete, consider the CleanUp environment, where the land is jointly owned by all agents and they work collectively for the prosperity of the society. The reward collected at each time step is shared equally among all the agents that work towards joint prosperity. This approach may however encourage the behavior of free-loading agents (i.e. agents might learn to not perform any task and still be rewarded because other agents are doing a good job and getting rewards that are shared with everyone.)

### 5.3 Anarchy

Anarchy is the state of a society being free of any authority or governing body. We consider the vanilla A3C model in the Cleanup environment as Anarchy, because all agents are free to do anything they wish and they get rewarded individually for their actions. It can cause the to agents greedily chooses to maximize their own gains without the concern for the society leading to sub-optimal outcomes for the whole community.

## 6 EXPERIMENTS AND RESULTS

To compare the effectiveness of the different economic paradigms mentioned in Section 5 for SSD problems, we use the Cleanup environment [1] depicted in Figure 1. We first define how different parameters for the experiments were chosen, followed by the main experiment, where we run the CleanUp game for the four different paradigms for two hardness levels of the environment. We also analyze the effects of pre-training agents to perform specific roles

[1]The code of experiments is available at https://bitbucket.org/alyibrah/capitalism/

and finally we discuss the social fairness of our approach. Socialism and Anarchy paradigms represent our baselines.

## 6.1 Selection of Hyper-parameters

The experiments are very compute intensive, therefore it was not feasible for us to run an extensive hyper-parameter search for the entire space. We ran the experiments for different values of hyper-parameters, changing one parameter at a time and keeping the rest fixed. The details of the run and the default hyper-parameters used can be seen in the Hyper Parameter Tuning Results appendix (A) at the end of the paper (Figure 5 and Table 2). The hyper-parameters we considered are :

- *Window Size ($w$)* : The window size is the number of steps in the past we consider cleaning work to be worth getting paid on. This is discounted to emphasize recent work. $W_{a,t} = \sum_{\tau=0}^{w} \gamma^{\tau} C_{a,t-\tau}$, where $C_{a,t}$ was the number of waste cells cleaned by agent $a$ in timestep $t$. We observe that the rewards are the least for window size 1 as delayed rewards are not accounted for. Very large values of window size also doesn't perform too well as it can make cleaning agents slack off since at some point they get reward irrespective of the action they pick in the recent time.
- *Discount Factor ($\gamma$)* : Having a very high discount factor (we treat all previous cleaning work equally valuable) has highest variance (instability) and doesn't take long to degrade the reward. It could be because the cleaners learned the unwanted lesson that if they clean for a number iterations (say $x$) they will take the yield of this for the next ($1000 - x$) iterations.
- *Dynamically Changing Apple and Dirt Spawn Rates* : In this experiment, we were trying to achieve specialization without implementing pre-training in Capitalism. To do this we decided to start the first iteration with a high apple spawn rate and a low waste spawn rate (aka. a very wealthy environment), then we decrease the former and increase the latter with each 30 iterations, till we reach the more stringent environment. What we found was that as soon as the rates reach the more stringent values, the learned policy does not help, and the aggregate rewards decrease substantially.
- *Ratio of Harvesters to Cleaners* : We observe in general that having more cleaners than harvesters yields higher rewards, as the dirt spawns at a much higher rate than the apple spawns. For the experiments we chose the ratio of cleaners to harvesters to be 3:2 instead of 4:1 for all paradigm even though it performs better, as we wanted at least two agents of each type.
- *Reward Redistribution Ratio ($\alpha$)* : This value is equivalent to the opposite of taxation, i.e. the portion of reward obtained by the harvester who collected the apple. Higher values of $\alpha$ don't work well as cleaners are not well compensated for their hard work, whereas very low values also doesn't incentivize the harvesters to do the collection work. Therefore a balanced value of $\alpha$ keeps both types of agents behave as per expectation.
- *Reward for Primary ($\beta$) and Secondary ($\epsilon$) Actions* : Recall that $\beta$ is the artificial reward for the primary action of the agent (e.g. cleaning for cleaners), and $\epsilon$ is the artificial reward for the secondary action for each agent (e.g. harvesting for cleaners) during the pre-training phase. A higher beta value should increase the amount of bias or specialization achieved by agents. We are using conservation of rewards in this experiment in the sense that $\epsilon = 1 - \beta$. A higher value of $\beta$ ensures that the agent performs the specialised task after the pre-training phase is over. A lower value makes the agents more flexible and they switch roles later on more frequently.

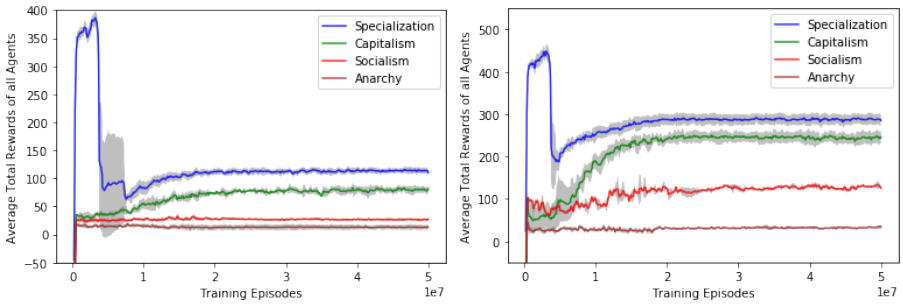
For the experiments for the Specialization paradigm, we observe that our approach is not very sensitive to the hyper-parameters selected.

## 6.2 Comparison of Different Paradigms

Using the hyper parameters in Table 2, next we perform 5 runs for the experiments of the four paradigms. Each run consists of 50k episodes and each episode was 1k iterations. The first 4k episodes for the specialization roles is the pre-training stage, where the agents are trained to specialize in their respective roles, therefore we see a bump in the rewards as it contains 'artificial rewards' for performing the cleaning task. *It is important to note here that this bump in rewards does not on its own give advantage to Specialization, because the environment resets randomly with each episode.* We also compare the results for an easier environment by changing the probability of the dirt and apple spawn rates. As seen in Figure 3, Capitalism with pre-training (Specialization) significantly outperforms all other methods: Capitalism, Socialism, and Anarchy. By performing pre-training, we bias the agents to specialize in a certain task. The reward redistribution in Capitalism further encourages this splitting and specialization of tasks. In Capitalism, rewards are redistributed in proportional to the amount of cleaning (social work) done in the given window, which encourages agents to continue their specialized behaviour and coordinate to achieve better rewards in the environment. By comparison, in Anarchy model, cleaning does not produce any rewards, therefore agents might not be encouraged to perform cleaning and thus the rate of spawn of apples decreases in the environment, which leads to poor overall rewards. In Socialism, rewards are equally redistributed among agents irrespective of the work done by agents, which can encourage the behavior of free-loading agents—agents might learn to not perform any task and still be rewarded (see Section 6.4). Therefore, we can intuitively reason as well as empirically observe how our approach performs better than Socialism and Anarchy on the same environment initialization.

## 6.3 Analysis of Pre-training Agents

We wanted to analyze the retention of specialization roles for the agents and contrast it with unspecialized agents in vanilla Capitalism, Socialism, and Anarchy. We compared the cleaning work and apple collection work for each agent for the different paradigms. In Figure 4, we plot for each paradigm the ratio of $\frac{\text{clean work}}{\text{total work}}$ for each agent. If we look at graph (a), we observe in the first 4k iterations that agents 1-3 are specializing in cleaning, while 4-5 are specializing in apple collection. Afterward, we see that agents 2 and 3 (both cleaners) have a higher cleaning ratio than agents 4 and 5 (both harvesters), while agent 1's (cleaner) cleaning ratio is lower relative to cleaner agents 2 and 3 because it is doing more harvesting work




**(a) apple & dirt spawn rate=(0.05,0.5)** **(b) apple & dirt spawn rate=(0.3,0.3)**

**Figure 3: Average overall agents rewards for paradigms with different apple & dirt spawn rate across five independent runs**

as compared to them (see Figure 6 in appendix (B)). It is as if agent 1 figured out that harvesting brings higher reward. While this is very promising, we cannot claim that this is a significant enough difference to indicate that specialization is retained. In the other paradigms, Figure (4[b-d]), no indication of specialization is observed. In Anarchy, agent 3 is not specializing, it is doing less work than rest of the agents. In Figure 6(a) in appendix (B), we see that the cleaner agents 2 and 1 are doing, on average, better cleaning work than the harvester agent 4. However, agent 5 (harvester) is doing far more cleaning work than agents 3 (cleaner). In (b), we see a notable difference in apple collection work between the cleaners 2-3 and harvesters 4-5. We conclude that although all agents are doing great collection work and cleaning work, *specialization* makes agents better learners (just like education makes people better equipped to create value), so agents sometimes do not stick to their majors. We believe that changing the hyper-parameters (e.g. taxation rate) could tilt the agents to stick to their specialization, which would be interesting to investigate.

## 6.4 Analysis of the fairness of the Algorithms

We employ a fairness metric to measure equity under our capitalistic constraints, and compare that to the other economic paradigms. We define utility $u_t^i = r_t^i - (c_t^i + h_t^i) \times \bar{r}$ where $r_t^i$ is the total reward of agent $i$ at time $t$. Let $c_t^i$ and $h_t^i$ be the amount of cleaning and harvesting work done respectively by each agent at time $t$. Smoothing on $u_t^i$ was done with a sliding window size of 2% the total number of timesteps. Denote by $\bar{r} = \frac{\sum_{i,t} r_t^i}{\sum_{i,t} c_t^i + h_t^i}$ the average reward per unit work (both cleaning and harvesting) among all agents and all timesteps for this paradigm, and let $\bar{u}$ be the average over $u_t^i$ across all agents at each timestep. This assumes cleaning and harvesting work are of equal value to the community. We use the non-negative Coefficient of Variation, $C_t$, (Equation (1)) introduced by previous work on Fairness in Multi Agent Games [9]. This measures whether agents get equal reward for equal work over each episode. A lower value here, corresponds to more equal $u_t^i$ for all agents. A higher value means agents are either free-riding, or not getting paid enough. In Table 1, we observe that the median value under the socialism paradigm, due to collective reward sharing irrespective of the work done (free-riding), the coefficient is high with wider IQR. For the anarchy paradigm, agents are not compensated for cleaning work, hence they have the highest co-efficient, indicating it is the least fair paradigm. Our proposed capitalistic reward shaping and the specialization variant, despite having no explicit equality in reward distribution, are more fair than our baselines. Specialization is more fair than capitalism due to a small amount of unfairness spikes in capitalism. These spikes are due to timesteps where no apples were collected and so the reward of cleaners was zero according to our formulation for $r_{a,t}$ in section 5.1. In capitalism, agents were not specialized, hence, each oscillated between harvesting and cleaning (see Figure 4 graph (b)), which led to fewer cleaning work being done (2.5 agents on average in capitalism as opposed to 3 agents in specialization), this caused intermittent scarcity of apples. In socialism and anarchy the overall rewards (welfare) is small (recall Figure 3), which is the unfortunate case in poor countries where an agent's (1) work is not adequately compensated, or (2) their reward is unfairly distributed.

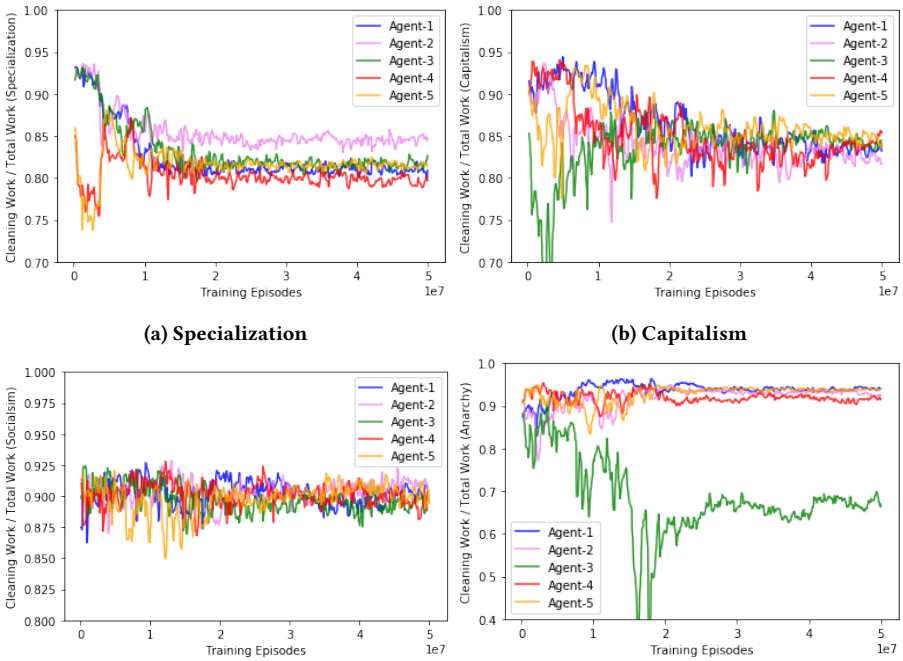
$$C_t = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} \frac{\left(u_t^i - \bar{u}\right)^2}{\bar{u}^2}} \tag{1}$$

**Table 1: Fairness of the Algorithms Proposed**

| Paradigm | Median | Inter-Quartile Range (IQR) |
|---|---|---|
| **Specialization** | 0.26960 | 0.07455 |
| **Capitalism** | 1.00289 | 1.00263 |
| **Socialism** | 1.16838 | 1.52340 |
| **Anarchy** | 17.86422 | 11.89171 |

## 7 DISCUSSIONS

Although Specialization outperformed other paradigms in this environment, we believe understanding why cleaning agent 1 did not retain their education/specialization (Figures 4 and 6) and what hyper-parameters should be tweaked to increase the specialization is a promising direction. It is also natural to ask whether agents actually specializing would yield a better overall reward or not in this case (or if adding specialization to our baselines might drastically improve their reward, our preliminary experiments show that it does improve rewards, but not enough to compete with Capitalism).




**(a) Specialization** **(b) Capitalism**

**(c) Socialism** **(d) Anarchy**

**Figure 4: Ratio of cleaning work done to total work done for each agent for different paradigms with apple & dirt spawn rates=(0.05,0.5) across five independent runs. Agents [1-3] are cleaners and Agents [4-5] are harvesters**

One possible explanation for the agent's behaviour in the Specialization paradigm, is it forgot its exact specialization, but retained that harvesting is somehow a better career.

We believe that our extrinsic motivation approach is a first step to formulate a critical paradigm for reward redistribution. Moreover, we maintain that this approach not only increases stability, but also equity among agents as seen in Table 1. We believe that our approach is robust to the underlying learning algorithm being used, as we only make changes in how the reward is being distributed to agents. We would need to validate our claims by running further experiments for other learning algorithms such as distributed Q-learning [18] or MADDPG [14]. Moreover, decreasing learning rate or constraining the policy after pre-training stage might prove to further improve the results of specialization.

This framework is general and should be applicable to any environment satisfying the properties of SSD as described in Section 2.1. It would be interesting to apply this reward redistribution framework on other SSD frameworks such as Harvest [6] (common dilemmas game) and others in the future.

## 8 FUTURE WORK

As part of the future work, we would like to test our algorithm's robustness to the changing dynamics of the environment (e.g. changing the location of the river, the apple spawn areas, the spawn probabilities of apples / dirt over time), which could be seen as an instance of transfer learning or continuous learning. Another set of experiments could be to change the taxed rewards based on the number of apples currently present, i.e. if the apples are abundant in the environment, then the harvesters should be tax less and vice versa. We would also like to ensure that our formulation maintains its advantage when increasing the population (number of agents) and indeed scales without any computational issues.

It would be interesting to try out variants of [8] where notion of trust among agents is established, such that agents are able to follow their influencers and even learn a mechanism of voting to elect them (as in democratic settings). Allowing to change the cognitive abilities of some agents (for instance, varying the agents' computational power, their learning capacity, or their partially observable state) would be very interesting to study. Following a realistic model of the world, a future direction could be seeing how our paradigms can be improved to be resilient to adversarial agents and devise strategies for agents to detect and attenuate their harmful effects.

## ACKNOWLEDGMENTS

We would like to thank Nicholas Feller and Google for giving us Cloud Compute Credits to run the hyper-parameter tuning experiments, and the Media Lab for the compute to run the rest of our experiments. We would also love to thank Jayakumar Subramanian for checking our initial thought process and guiding us to some of the literature covering cooperative MARL.

# REFERENCES


[1] Kris Cao, Angeliki Lazaridou, Marc Lanctot, Joel Z Leibo, Karl Tuyls, and Stephen Clark. 2018. Emergent communication through negotiation. *arXiv preprint arXiv:1804.03980* (2018).

[2] Nuttapong Chentanez, Andrew G Barto, and Satinder P Singh. 2005. Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*. 1281–1288.

[3] Tom Eccles, Edward Hughes, János Kramár, Steven Wheelwright, and Joel Z Leibo. 2019. Learning Reciprocity in Complex Sequential Social Dilemmas. *arXiv preprint arXiv:1903.08082* (2019).

[4] Samah El-Tantawy and Baher Abdulhai. 2010. An agent-based learning towards decentralized and coordinated traffic signal control. In *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 665–670.

[5] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in neural information processing systems*. 2137–2145.

[6] Edward Hughes, Joel Z Leibo, Matthew Phillips, Karl Tuyls, Edgar Dueñez-Guzman, Antonio García Castañeda, Iain Dunning, Tina Zhu, Kevin McKee, Raphael Koster, et al. 2018. Inequity aversion improves cooperation in intertemporal social dilemmas. In *Advances in Neural Information Processing Systems*. 3326–3336.

[7] Maximilian Hüttenrauch, Sosic Adrian, Gerhard Neumann, et al. 2019. Deep reinforcement learning for swarm systems. *Journal of Machine Learning Research* 20, 54 (2019), 1–31.

[8] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro A Ortega, DJ Strouse, Joel Z Leibo, and Nando de Freitas. 2018. Intrinsic social motivation via causal influence in multi-agent RL. *arXiv preprint arXiv:1810.08647* (2018).

[9] Jiechuan Jiang and Zongqing Lu. 2019. Learning Fairness in Multi-Agent Systems. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 13854–13865. http://papers.nips.cc/paper/9537-learning-fairness-in-multi-agent-systems.pdf

[10] Wei Jiang, Gang Feng, Shuang Qin, Tak Shing Peter Yum, and Guohong Cao. 2019. Multi-agent reinforcement learning for efficient content caching in mobile D2D networks. *IEEE Transactions on Wireless Communications* 18, 3 (2019), 1610–1622.

[11] Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 2018. Emergence of linguistic communication from referential games with symbolic and pixel input. *arXiv preprint arXiv:1804.03984* (2018).

[12] Joel Z Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. 2017. Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 464–473.

[13] Xuedong Liang, Ilangko Balasingham, and Sang-Seon Byun. 2008. A multi-agent reinforcement learning based routing protocol for wireless sensor networks. In *2008 IEEE International Symposium on Wireless Communication Systems*. IEEE, 552–557.

[14] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*. 6379–6390.

[15] Michael W Macy and Andreas Flache. 2002. Learning dynamics in social dilemmas. *Proceedings of the National Academy of Sciences* 99, suppl 3 (2002), 7229–7236.

[16] Laëtitia Matignon, Laurent Jeanpierre, and Abdel-Illah Mouaddib. 2012. Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes. In *Twenty-sixth AAAI conference on artificial intelligence*.

[17] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*. 1928–1937.

[18] Hao Yi Ong, Kevin Chavez, and Augustus Hong. 2015. Distributed deep Q-learning. *arXiv preprint arXiv:1508.04186* (2015).

[19] Liviu Panait and Sean Luke. 2005. Cooperative multi-agent learning: The state of the art. *Autonomous agents and multi-agent systems* 11, 3 (2005), 387–434.

[20] Peng Peng, Quan Yuan, Ying Wen, Yaodong Yang, Zhenkun Tang, Haitao Long, and Jun Wang. 2017. Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069* 2 (2017).

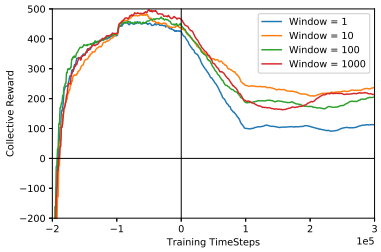
[21] Anatol Rapoport. 2012. *Game theory as a theory of conflict resolution*. Vol. 2. Springer Science & Business Media.

[22] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. 2016. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295* (2016).

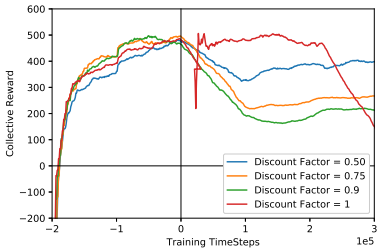
[23] MA Wiering. 2000. Multi-agent reinforcement learning for traffic light control. In *Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000)*. 1151–1158.

[24] Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Başar. 2018. Fully decentralized multi-agent reinforcement learning with networked agents. *arXiv preprint arXiv:1802.08757* (2018).

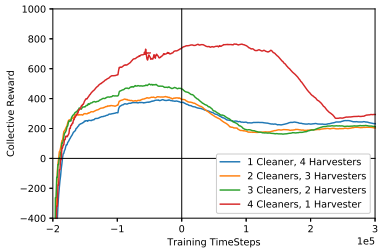
# A HYPER PARAMETER TUNING RESULTS



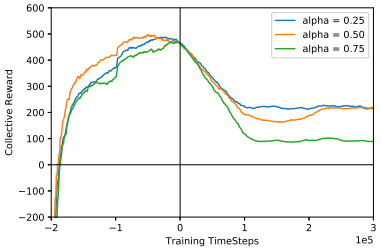

(a) Varying the Window Size



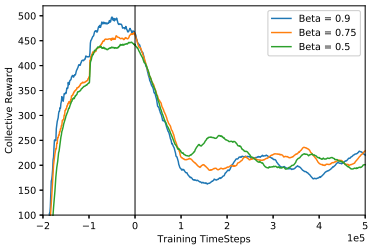

(b) Varying the Discounting Factor




(c) Varying the Number of Harvesters and Cleaners




(d) Varying the Redistribution Ratio



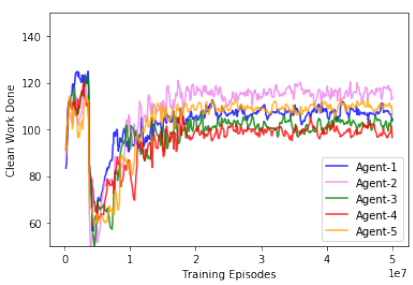

(e) Varying the Reward for the Primary Action

**Figure 5: Figures [a-e] show the overall rewards of all agents upon varying different hyper-parameters for Specialization paradigm. The experiments were run for 700 episodes with 1000 iterations per episode. The first 200 iterations are the specialization stage, therefore we see a bump in the rewards. For all the above experiments, we have $\alpha = 0.5$, $\beta = 0.9$, $\epsilon = 0.1$, *cleaners* $= 3$, *harvesters* $= 2$, *apple & dirt spawn rate* $= (0.3, 0.3)$, and *windowsize* $= 1000$ as the default values. When varying a specific hyperparameter we keep the others constant at these values.**

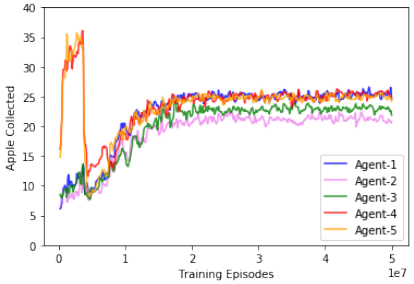
**Table 2: The ideal hyper-parameters selected for capitalism and specialization paradigms after running 700 episodes.**

| Paradigm | $\beta$ | $\epsilon$ | window-size | discounting | harvesters | cleaners | $\alpha$ |
|---|---|---|---|---|---|---|---|
| Specialization | 0.9 | 0.1 | 10 | 0.5 | 2 | 3 | 0.5 |
| Capitalism | N/A | N/A | 100 | 0.9 | 2 | 3 | 0.5 |

# B ANALYZING RETENTION OF SPECIALIZATION FOR AGENTS




(a) Agent-wise clean work




(b) Agent-wise apple collected

**Figure 6: Total dirt blocks cleaned and total apples collected by each agent per episode averaged across 5 runs for specialization paradigm with the apple & dirt spawn rates of (0.05,0.5)**