Collective eXplainable AI: Explaining Cooperative strategies and agent contribution in Multiagent Reinforcement Learning with Shapley Values

Alexandre Heuillet[§], Fabien Couthouis[§] and Natalia Díaz-Rodríguez

***Abstract*—While Explainable Artificial Intelligence (XAI) is expanding more and more areas of application, little has been applied to make deep Reinforcement Learning (RL) more explainable. As RL becomes ubiquitous and used in critical and general public applications, it is essential to develop methods to make it better understood and more interpretable. In this study, we propose a novel approach to explain cooperative strategies in multiagent RL using Shapley values, a game theory concept that successfully explains some Machine Learning algorithms. We argue that Shapley values are a pertinent way to evaluate the contribution of players in a cooperative multi-agent RL context. To palliate the high overhead of this method, we approximate Shapley values using Monte Carlo sampling and evaluate this method on two cooperation-centered and socially challenging multi-agent environments (*Multiagent Particle* and *The Commons' Sequential Social Dilemmas*). We show that Shapley values succeed at estimating the contribution of each agent. Moreover, this work could have implications that go beyond games in economics science (e.g. in nondiscriminatory decision making or policy making under fairness constraints). However, Shapley values only give general explanations about a model and cannot explain a single run, episode nor justify precise actions taken by agents. We advocate for these needs for future work to address these critical aspects.**

***Index Terms*—Reinforcement Learning, Explainable Artificial Intelligence, Responsible Artificial Intelligence, Shapley values**

I. Introduction

Over the last few years, Reinforcement Learning (RL) has been a very active research field. Many RL-related works focused on improving performance and scaling capabilities by introducing new algorithms and optimizers (e.g. [1]–[3]) whereas very few tackled the issue of explainability in RL [4], [5].

However, explainability in Machine Learning (ML) is becoming more and more a pressing issue as it concerns general public trust, and the transparency of algorithms now conditions the deployment of ML, and thus RL, in industry and daily life. As a consequence, Explainable Artificial Intelligence (XAI) was designed as a new field of study that strives to bring explainability to every ML aspect such as linear classifiers [6], time series predictors [7], [8] or RL [9]–[11]. In fact, even if existing works managed to provide explanations for specific situations [9]–[11], RL models still lack a general explainability framework similar to SHAP [12] or LIME [6], designed for ML, that would allow a broader form of explainability for RL.

Thus, in this work, inspired by SHAP [12], we explore the possibilities offered by the mathematical framework of Shapley values [13] to explain RL models, with a focus on multi-agent cooperative environments, also called "Common Games". They are challenging scenarios where all participants must cooperate in order to achieve a common goal. In the particular case where agents must gather resources from a common pool without being greedy, Garrett Hardin [14] defined what he called "The Tragedy of the Commons": if an agent uses slightly more resources than it should, this might be inconsequential, but if every agent starts following this logic then the consequences can become dire with the common pool being exhausted and no one being able to gather resources anymore. This is why these settings are especially interesting to tackle using RL while finding ways to evaluate and understand how agents cooperate and share resources using explainability methods such as Shapley values.

As a matter of fact, this work could have implications that go beyond games, since RL-based systems are increasingly used to solve critical problems. In particular, the relevance of studying social dilemmas and explaining the contribution of each policy [15], agent, or each model feature becomes relevant in many societal problems. For instance, it could provide useful insights in economic science of social structures, allocating resources or designing resilience programs (e.g. climate change, or in nondiscriminatory decision making or policy design for fairness). Moreover, in the case of COVID-19 pandemic policies or collective risk dilemmas, it is possible to study the effect of tourism restriction policies on economic recovery based on a multiagent evolutionary game model.

Our main hypothesis is that Shapley values can be a pertinent way to explain the contributions of agents in multi-agent RL cooperative settings and that valuable insights, especially for RL developers, can be derived from their analysis. We present experiments (see Section V) conducted on Multiagent Particle [16] and Sequential Social Dilemmas [17] environments, and show that Shapley values can accurately answer the following research questions:

- *Can Shapley values be used to determine how much each agent contributes to the global reward? (RQ1)* If the

A. Heuillet is with IBISC, Université Paris-Saclay, Univ Evry, 36 rue du Pelvoux, 91020 Evry-Courcouronnes, France e-mail: alexandre.heuillet@universite-paris-saclay.fr.

F. Couthouis is with Ubisoft Player Analytics France, Ubisoft Paris Studio, 66 - 72 Rue Marceau, 93100 Montreuil, France email: fabien.couthouis@ubisoft.com.

N. Díaz-Rodríguez is with Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI Institute), University of Granada, Spain. email: natalia.diaz@ensta-paris.fr

[§]Equal contribution

answer to RQ1 is yes:

- *Does the proposed Monte-Carlo based algorithm empirically offer a good approximation of Shapley values? (RQ2)*
- *What is the best method to replace an agent missing from the coalition (e.g. random action, action chosen randomly from another player or the "no operation" action)? (RQ3)*

Our experiment set up is brought forward as well in order to show the limitations of the Shapley framework within explainable RL. We illustrate how this approach, on the contrary, cannot explain particular notions of a multi-agent learning model. In particular, explaining the contribution of a specific episode or a specific action taken by an agent, at a given point in time of its training. This is due to the requirements that come along with training multi-agent RL, and the design limitations of the Shapley framework, which we bring upfront. In particular, we show (as discussed in Section IV), how Shapley values only yield an average metric of each player contribution to the overall reward and thus, to obtain this average contribution metric, one must compare the cooperation of players during several games (or episodes in RL). However, due to the stochasticity of the environment as well as the non-deterministic behaviour of different agents, identical conditions –non inherent to the agent's policy at consideration– make concrete episodes, or concrete actions within an episode, not comparable.

This article presents the following contributions:

- A study of the mathematical notion of Shapley values and how it is able to provide quantitative explanations about the individual contribution of agents in a cooperative multi-agent RL environment.
- A global model-agnostic method to explain multi-agent cooperative RL models using Monte Carlo approximated Shapley values.
- A set of experiments that demonstrates the applicability and usefulness of Shapley values and how they can be very insightful in an explainable RL multi-agent cooperative setting.

The rest of this article is structured as follows: Section III presents some preliminaries about RL and Shapley values, Section VI discusses the experimental study described in this article and the general usage of Shapley values in a RL setting and finally Section VII brings a conclusion on what experiments carried out convey in terms of explainable RL, and gives some insights on promising lines of future work.

II. Related work

Cooperative multi-agent RL has been studied in different settings, e.g. the emergence of different behaviours has been studied in the context of the Commons Tragedy game [18]. For instance, showing that certain inequity aversion improves intertemporal social dilemmas [19]. However, these studies analyze the game from a theoretical point of view and not from the XAI angle. We are particularly interested in explaining the most relevant factors (be it an agent, episode, policy, action) that contributes the most within a black box (deep RL) model to learn a particular policy.

Recent studies about eXplainable Reinforcement Learning (or XRL [5]) can be categorized in two main categories mainly designed within the XAI literature [20], [21] transparent methods and post-hoc explainability (according to the XAI taxonomies in [21]). On the one hand, transparent algorithms include by definition every ML algorithm that is understandable by itself, such as a decision-tree. On the other hand, post-hoc explainability includes all the methods that provide the explanation of an RL algorithm after its training, such as LIME [6] or SHAP [12] for standard ML models. In a ML context, SHAP (SHapley Additive exPlanations) [12] is a post-hoc explainability method that is able to explain causal relations between the inputs and outputs of any trained model using Shapley values [13], a notion borrowed from game theory. Shapley values are discussed in detail in Section III-B and its limits at explaining ML models are explored by Camburu et al. [22]. Numerous other studies [23]–[25] tried to transpose Shapley values into XAI but, to the best of our knowledge, none tried to apply this method to the specific issue of cooperative multi-agent RL in the context of attaining explainable XRL. In fact, most XRL methods are based on transparent algorithms [5].

However, recent works also tried to explain multi-agent RL. Wang et al. [26] developed an approach named Shapley Q-values to solve global reward games in a multi-agent context, based on Shapley values and DDPG (Deep Deterministic Policy Gradient). Their method relies on distributing the global reward more efficiently across all agents. Indeed, integrating Shapley values into DDPG enables sharing the global reward between all agents as a function of their contributions. The more the agent contributes, the more reward it will get. This contrasts to the equally shared reward approach, which could cause inefficient learning by giving rewards to an agent who contributed poorly. Their experiments showed that SQDDPG (Shapley Q-value DDPG) presents faster convergence rate and fairer credit assignments [27], in comparison with other algorithms (i.e. IA2C, IDDPG, COMA and MADDPG (Multiagent DDPG)). This method allows to plot the credit assignment to each agent, which can explain how the global reward is divided during training, and what agent contributed the most to get the global reward.

All methods listed above apply Shapley values in a ML context by considering model features as participants of a cooperative multiplayer game. In this article, we apply it to a cooperative multi-agent RL context by considering agents as players instead, an approach closer to the original game theory method presented by Lloyd Shapley [13].

However, one limitation of this game theory approach (Shapley values) is that it does not allow to obtain the contribution of one agent on a specific learning episode. Other methods present in the literature take a different focus. For instance, *BreakDown* [28] is another interpretable approach applied to predictive modeling that shows the contributions of each feature to the prediction, but computes them step by step: it starts with an empty team, adds the feature value that would contribute the most to the prediction and iterates until all feature values are added. Although faster than the SHAP [12] value method, for models without interactions (i.e. offline)

it provides the same results.

We saw some strategies to make Shapley values computation more bearable. Some consist of limiting the number of iterations (through reducing the evaluated coalitions) using Monte-Carlo approximation (see Section IV), or by following the greedy approach to compute a single series of nested feature conditionings. While the classical Shapley value approach computes the contribution of a single feature averaged across all possible conditionings, these computationally cheaper alternatives reduce the computation needs at the cost of increasing the variance of the Shapley values estimation.

While exact computation methods for both BreakDown and SHAP exist only for linear regression and tree ensemble models, the fact of depending on the number of samples of subsets of predictors p to be used makes, for more complex models, the estimated contributions by both methods to be different and potentially point in opposite directions [28].

In relation to interaction among features in cooperative games, Gosiewska et al. [29] argue that general interpretability frameworks that rely on additive explanations such as BreakDown or SHAP may sometimes be misguiding. In particular, when the model to explain lacks additive explanations, both methods will generate inconsistent explanations and omit large parts of the model behavior. This indicates that these generic methods are not the universal response to XAI yet.

In this article, despite these statements about general XRL frameworks, we show that in a cooperative multi-agent RL setting, Shapley values can be used to accurately evaluate the contributions of different agents.

III. Explainability and Shapley Values for RL in cooperative settings

A. Explainability in cooperative RL

With the rapid growth of RL research and industrial applications (e.g. autonomous systems [30], [31], robotics [32], [33]) we have witnessed over the past few years, a need for Explainable RL (XRL) has risen as being able to understand and justify the decisions of such models is legally and morally necessary for their broad diffusion. Thus, it is a very young research field with only a small number of articles published to this day, as stated by recent surveys conducted on the matter [4], [5].

Furthermore, the subfield of XRL that focuses on multi-agent cooperative games is even more restricted but recently gained significant attention with emerging concepts such as social learning in a RL context. While studying social interactions between entities in cooperative games is originally a specificity of sociology or economics [34], [35], some RL researchers realized that they could potentially provide explanations or improve their models this way.

Perolat et al. [18] sought to conduct new behavioral experiments using RL agents in video-game like cooperative environments instead of human subjects. More accurately, they studied games which put the emphasis on common-pool resources (CPR) and found that agents learn behaviors that are emergent and that strategies where some agents are excluded from the CPR can arise. In addition, they came up with metrics that quantify social outcomes such as sustainability, equality or selfishness for RL models.

Following the same direction, Jaques et al. [36] proposed a framework to achieve better coordination and communication between agents by taking into account the causal influence (i.e. actions leading to big changes in the other agents' behavior) that some of them can have on others and rewarding them in consequence. Their empirical results show that agents that choose their actions carefully in order to influence the others lead to better coordination and thus better global performance in socially challenging settings where cooperation is crucial.

Exploring this aspect further, Ndousse et al. [37] analyzed the behavior of independent RL agents in multi-agent environments and found out that model-free agents do not use social learning. Thus, they introduce a model-base auxiliary loss that allows agents to learn from other well-performing other ones ("experts") to improve themselves. In addition to outperforming the experts, these agents were also able to achieve better zero-shot performance than those which did not rely on social learning when transferred to another task.

However, even if these works managed to extract useful information from studying social interactions between agents, the literature lacks a general framework that could automatically provide explanations about the level of performance of each agent and their added value in the cooperative game, such as SHAP [12] does for features of a ML model.

B. Shapley values and RL in cooperative settings

Shapley values originate from game theory models and they evaluate the importance in terms of contribution of each participant in a cooperative game, in order to help split in a fair way a shared payout [13]. The concept which is paramount here is to be able to form "coalitions" (or subsets) of players in order to measure the performance of each player in every possible team situation (for instance, "player A", "player A and player B" or "player B and player C").

Formally, a coalitional game $C = (N, v)$ is defined by a set N of players with $|N| = n$ (the number of players) and a function v, that maps a coalition of players S to a real number, corresponding to the total expected sum of payouts the members of S can obtain through cooperation: $v : 2^N \Rightarrow \mathbb{R}$, with $v(\varnothing) = 0$ and $\varnothing$ being the empty set. Thus, v is denominated the gain function of the considered game.

The idea is to quantify how much players cooperate in a coalition and receive a certain profit from this cooperation [38]. According to the Shapley value definition presented in [13], the contribution added by player i in a coalition S in a coalitional game (N, v) is given by Eq. 1:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} (v(S \cup \{i\}) - v(S)) \quad (1)$$

A more interpretable but equivalent formula to express the Shapley value for player i, rewritten with binomial coefficients [39], is:

$$\phi_i(v) = \frac{1}{n} \sum_{S \subseteq N \setminus \{i\}} \binom{n-1}{|S|}^{-1} (v(S \cup \{i\}) - v(S)) \quad (2)$$

In summary, the Shapley value of a feature (or *player*) is the mean marginal contribution (the term: $v(S \cup \{i\}) - v(S)$ in (2)) of all possible coalitions, or average change in the prediction that the coalition already in the room receives when the feature value joins them. It satisfies desirable properties [13], [40] lacking in other XAI techniques: *Efficiency* (the sum of the Shapley values of all players equals the value of the grand coalition), *Dummy, or dummy feature* [38], [40] (if a feature does not change the predicted value, e.g. the global reward in RL, regardless of which coalition of feature values it is added to, then its Shapley value is equal to zero), *Symmetry* (the contributions of two feature values should be the same if they contribute equally to all possible coalitions) and *Linearity* (the contribution of a coalition of features should be the sum of the individual contributions of features that compose the coalition).

It is important to note that the sum of Shapley values yields the difference of actual and average prediction. The correct interpretation is as follows: the Shapley value is the average contribution of a feature to the prediction in different coalitions. Note that it is not the difference in prediction when we would remove the feature from the model [38]. As shown in Eq. 1, for $|N| = n$ players, the exact computation of Shapley values for a specific participant requires computing the average of 2^{n-1} possible coalitions, which is computationally expensive, especially when considering all players. Therefore, it means computing $n(2^{n-1})$ coalitions in total to obtain values for all players. While this value can be reduced to 2^n coalitions (for all players, if we optimize the algorithm by avoiding computing the same coalition multiple times), the cost remains exponential w.r.t. the number of players. In fact, the number of agents in RL environments can vary greatly from a few [16], [41], [42] to hundreds [43], [44], where exact Shapley values are prohibitively expensive to compute. Furthermore, estimating Shapley values in a stochastic RL environment requires sampling multiple episodes in order to estimate all marginal contributions, which worsens the complexity.

In terms of computational efficiency, assuming that simulating a game is done in constant time $O(1)$, we can determine that the exact solution to this problem becomes difficult as the number of coalitions exponentially increases as more features are added. Despite Shapley value computation being an NP-hard problem [45], Shapley distributes the feature attribution fairly, i.e., allowing contrastive explanations, for instance, permitting the comparison of a prediction to a feature subset prediction, or a single data point.

In spite of the broad applicability of Shapley values, they have some theoretical limitations. In particular, Shapley values are only a way to obtain an average metric of each player's contribution to the overall reward (or shared payout when not in a RL context). In order to obtain this average metric one must compare the cooperation of players during several games (or episodes in RL). Thus, it obviously cannot explain a single match (or game/episode) contribution to the learning –i.e., a decisive match in the learning experience of a football player – nor explain one specific action taken at a single moment by a player.

IV. Monte-Carlo approximation of Shapley values

Since the complexity of computing Shapley values grows exponentially with the number of players (as discussed in Section III-B), in order to keep the calculation time manageable, we can compute contributions for only a subset of all possible coalitions. The Shapley value $\phi_i(v)$ can be approximated by Monte-Carlo sampling in order to apply it to any type of classification or regression model [46] as follows:

$$\hat{\phi_i}(\hat{f}) = \frac{1}{M} \sum_{m=1}^{M} (\hat{f}(x^m_{+i}) - \hat{f}(x^m_{-i})) \approx \phi_i(\hat{f}) \tag{3}$$

Where $\hat{f}(x^m_{+i})$ is the model prediction (or gain function in a game theory context) for input x with a random number of feature values replaced by feature values from a random data point, except for the respective value of feature i and M is the number of marginal contributions to estimate in order to compute the Shapley value for one feature. The value of x^m_{-i} is almost identical to the last one, but the value x^m_i is also taken from this randomly sampled data point.

In this work, our contribution consists of adapting Eq. 3 to the multi-agent RL setting, by replacing input features by agent actions. While in Eq. 3 $\hat{f}$ is the function approximated by a classification or regression model [46], in our work, we consider instead $\hat{f}$ to be the global reward obtained by the agents from a random subset (or coalition) on a sample episode, leading to the following reformulation of Shapley value:

$$\hat{\phi_i}^{RL}(r) = \frac{1}{M} \sum_{m=1}^{M} (r^m_{+i} - r^m_{-i}) \approx \phi_i(r) \tag{4}$$

where r^m_{+i} corresponds to the global reward obtained by simulating one sample episode with a random subset of players where player i is present, and r^m_{-i} is the global reward obtained by simulating one episode with the same subset than in r^m_{+i}, except that the current player i has been removed from the subset.

We explore three approaches to exclude players from a coalition (i.e., let the absent player take "substitute" actions):

a *Replace*: We replace a missing agent's actions by those of a randomly chosen player among the trained agents which are present in the coalition (and ideally with the same role as the missing one). This is the direct translation from the standard application of Shapley values in ML [12] (against the traditional use in game theory where it is often possible to completely remove a player) since they replace missing feature values by ones randomly selected among present (non-zero) features.

b *Random*: Letting the absent player act by taking random actions.

c *NoOp*: Replacing missing agent's actions by "noop" (no operation) ones, i.e. letting the agent do nothing (not move).

We repeat the estimation of the Shapley value for each player. Thus we have to rollout $2M$ times per player ($2Mn$ rollouts in total, where n is the total number of players in the game). At the end of the process, we obtain one Shapley value per agent policy, indicating each player average contribution to the grand

coalition global reward (i.e. the reward collectively obtained by all the agents working simultaneously) on the sampled episodes (Monte Carlo method, see Algorithm 1). Hence the complexity of this method only depends on M ($O(M)$) as we can notice in Table II. The Shapley value estimation [46] in Eq. 3 allows the model to conclude, for instance, *On average, Player 1's contribution to the team has an impact of +0.6 on the global reward"*. Finally, this allows us to know how relevant each player is to the overall cooperation. Algorithm 1 describes the algorithmic process to estimate Shapley values via Monte-Carlo sampling of coalitions of players:

Algorithm 1 Monte-Carlo approximation of Shapley values applied to a multi-agent RL context

Input: List: $agents$
Input: Integer: M (number of coalition permutations to be used)
Output: List: $shapley\_values$
$shapley\_values \leftarrow empty\_list()$
for $i \leftarrow 1$ to $length(agents)$ **do**
  $marginal\_contributions \leftarrow empty\_list()$
  for $m \leftarrow 1$ to M **do**
    $coalition\_with\_i \leftarrow sample\_coalition(agents[i])$
    $coalition\_without\_i \leftarrow remove\_from\_list(coalition\_with\_i, agents[i])$
    $r_{+i} \leftarrow rollout(coalition\_with\_i)$
    $r_{-i} \leftarrow rollout(coalition\_without\_i)$
    $add\_to\_list(marginal\_contributions, (r_{+i} - r_{-i}))$
  end for
  $shapley\_value\_i \leftarrow mean(marginal\_contributions)$
  $add\_to\_list(shapley\_values, shapley\_value\_i)$
end for
return $shapley\_values$

In the previous examples and in Shapley value estimation [46] in Eq. 3 we saw different adaptations to apply Shapley values as an XAI method. We may use the notion of contribution on classification/regression model predictions (f), but also on RL reward on sample episodes (r) as final *payout* we need to explain. The analogy could be easier to understand if we used value function as in [26] instead of rollouts to evaluate each player contribution. With this generic approach, the more a feature (or player) is important, the more its presence will lead to a higher reward on average. We can thus use the following *contribution ranking scheme*: Rank agents in order of average importance in the team (by making coalitions of agents to estimate marginal contributions on the mean final reward), as explained above. This is the approach we present in this paper.

V. Experimental study

A. Context and Hypotheses

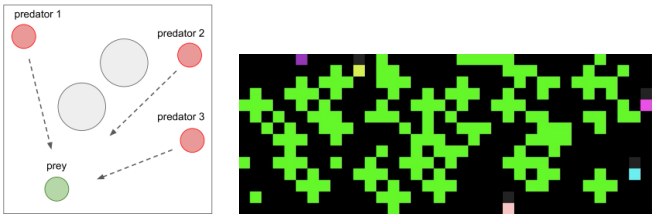
The straightforward application of Shapley values for making an AI model explainable consists of considering the features of a model as participants of a cooperative game, and the final prediction as the shared payout [12], [47], [48]. Computing the Shapley value of each feature allows to know its weight in the final decision. This idea has been used to explain ML models in SHAP [12].

Going further, we can make use of this attribution concept to get a post-hoc global explainability method for RL. One application to explain RL algorithms is to consider a multi-agent setup and use Shapley values to compute the contribution of each agent to the group's global reward. In this setup, the participants of the cooperative game will be the agents, and the shared payout is the global reward obtained by the agents at the end of an episode. By using the method described in Section IV to estimate the Shapley value for each agent, we will determine how much each agent contributes to the final reward in a set of experiments. Thus, we expect that each agent would obtain a Shapley value proportional to its contribution to the collaborative task and try to answer to the research questions defined in Section I (RQ1, RQ2, RQ3).

Multiple experiments were conducted [§] using two multi-agent RL environments and three different RL algorithms:

- *Multiagent Particle* (Predator Prey scenario) [16]: a light environment with a continuous observations and a discrete action space, along with some basic simulated physics and numerous scenarios. We chose to use the Predator-Prey scenario with three predators (represented as colored circles in Figure 1a) and a single prey (represented as a smaller light green circle). In this scenario, the prey is faster and wants to avoid being caught by predators. The collaborative task is thus for the predators to form a strategy to catch the prey, while avoiding the randomly placed obstacles (represented as black circles). The prey is positively rewarded when escaping the predators, and negatively rewarded when caught, and vice versa for predators. Both types of agents are rewarded negatively when trying to overpass the screen boundaries or hitting an obstacle. As it was done by the authors of Multiagent Particle [16], we trained the prey using the DDPG [49] algorithm [§] whereas predators were trained thanks to MADDPG [16][§]. According to [16] training preys with DDPG rather than MADDPG makes the challenge easier for the predators, thus allowing us to obtain good models (i.e. where the predators elaborates effective strategies to catch the preys) with little training time. For this scenario, we trained 5 different models (in order to obtain meaningful values despite the stochastic nature of MADDPG), with the same hyperparameters (as suggested in [50] and detailed in the Appendix). We also trained a model with independent PPO (IPPO) [51], in order to improve the experiments' generality.
- *Sequential Social Dilemmas* (Harvest scenario) [17]: another light environment with continuous observations and discrete actions that proposes scenarios that emphasize social interactions and cooperation between agents. We used an open-source implementation of this environment [52]. In the Harvest scenario, the agents must cooperate to harvest the maximum number of apples while being careful to not "kill" apple trees by collecting all the apples they contain, as this would prevent this tree from spawning further apples (see consecutive blocks forming apple trees in the environment in Figure 1b). As it was done by the authors of Sequential Social Dilemmas [52], all agents

[§]The repository linking to our experiments can be found here: https://github.com/Fabien-Couthouis/XAI-in-RL
[§]DDPG implementation from repository https://github.com/openai/maddpg/
[§]Used MADDPG implementation: https://github.com/openai/maddpg/



(a) (b)

Figure 1: (a) Predator-Prey scenario of Multiagent Particle environment [16]. (b) Screenshot of the *Harvest* scenario of Sequential Social Dilemmas environment [52]. Green blocks represent apples, while colored blocks are agents whose goal is to collect apples without tarnishing their source. Consecutive chains of apples are trees whose role is to regrow apples at a rate that depends on the number of nearby apples.

were trained using the Asynchronous Advantage Actor-Critic (A3C) [1][§] algorithm. We used the best reported hyperparameters and training protocol as reported in [36], [52].

We conducted two experiments on the Predators-Prey scenario: first one with the default settings provided by authors of the environment [16] (to verify RQ1, RQ2 and RQ3) and then one with different speeds for each predator (to further confirm RQ1 and RQ2). Speeds used on each experiment are presented in Table I. For *Harvest*, we also conducted three experiments: first we performed Shapley Values computation of a simple model trained with 6 agents and the default settings suggested by [36], [52] to further investigate RQ1 and especially the minimal number of agents required for this task. Then we ran another computation using the same model but modified according to information extracted from the Shapley analysis obtained during the first experiment, in order to confirm the validity of such information. In addition, we reported the social outcome metrics from [18] to have a more fine-grained view of the payout notion (instead of merely a global reward). We implemented the following metrics:

- *Efficiency* (Eq. 5): measures the total sum of all rewards obtained by all agents.
- *Equality* (Eq. 6): measures the statistical dispersion intended to represent inequality (Gini coefficient [53]).
- *Sustainability* (Eq. 7): defined as the average time $t_s \in t$ at which the rewards are collected.

Considering N independent agents, let $\{r_t^i | t = 1, ..., T\}$ be the sequence of rewards obtained by the i-th agent over an episode of duration T timesteps. Its return is given by $R^i = \sum_{t=1}^{T} r_t^i$. Thus, the equations describing the social metrics are as follows:

$$\text{Efficiency U} = \mathbb{E}[\frac{\sum_{i=1}^{N} R^i}{T}] \tag{5}$$

$$\text{Equality E} = 1 - \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}|R^i - R^j|}{2N\sum_{i=1}^{N} R^i} \tag{6}$$

[§]A3C implementation from repository https://github.com/ray-project/ray/tree/master/rllib

$$\text{Sustainability S} = \mathbb{E}[\frac{1}{N}\sum_{i=1}^{N} t^i], \text{where } t^i = \mathbb{E}[t | r_t^i > 0] \tag{7}$$

We finally executed additional experiments on different *Harvest* checkpoints to explore how the agents cooperate. We will detail each experiment in next subsections.

	Prey	Pred. 1 (slow)	Pred. 2 (medium)	Pred. 3 (fast)
Speed in exp. #1	1.3	1.0	1.0	1.0
Speed in exp. #2	1.3	0.2	0.8	2.0

Table I: Settings used for Experiment 1 and 2 (RQ1, RQ2, RQ3), conducted on the Predator-Prey setting of Multiagent Particle [16] (speed of each agent).

M	Computation time
100	$\approx$ 1 hour
500	$\approx$ 5 hours
1000	$\approx$ 10 hours

Table II: Table reporting Shapley values computation time for the Harvest [52] experiments (5 agents) run on a 6-core AMD Ryzen 5 5600X CPU, and using Algorithm 1. Computation time grows proportionally to M.

B. Experiment 1: Agents with identical settings in Multiagent Particle

1) Environment Settings: The goal of this experiment is to check if RQ1 and RQ2 (detailed in Section I) are valid (i.e. if the Shapley values of the predators are the same, corroborate with the number of times they catch the prey and are a close approximation of the exact Shapley values). First, when training a model of 3 predators using the default settings on the Prey-Predators scenario, we could think that each predator should provide a similar contribution as the predators do not have significant differences in their speed, action space or training method. However, in contrast with this assumption, statistics presented in Figure 2 show that the performance of each predator agent (i.e. the number of times each predator catch the prey) varies significantly as we can see that Predator 3 has a higher contribution than Predator 1, which have a higher contribution than Predator 0. In fact, the trained MADDPG model has developed a strategy in which Predator 3 and Predator 2 perform better than Predator 0 at catching the prey. This can be explained by the fact that MADDPG provides the same reward to all agents, instead of only rewarding the agent which contributed the most, which can bias the training, as explained in [26]. On the contrary, rewarding only the agent who contributed the most does not highlight and value the team strategies where the contribution of every agent has made the action possible.

2) Shapley values analysis: Figure 3 shows the Shapley values computed for each agent on each of the five models, over 1,000 sample episodes per model. This can be observed in a more convenient way in Figure 2, which shows Shapley values computed for each agent on a single model. As hypothesized, the order of agents contributions is the following: Predator 2, Predator 1 and Predator 0. Thus, this first experiment supports RQ1 as Shapley values are able to map correctly contributions to agents in a cooperative multi-agent setting.

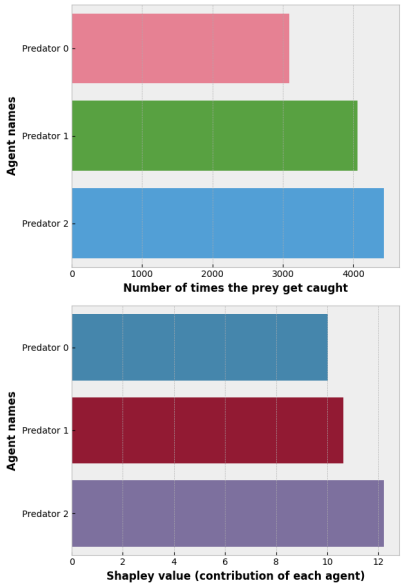


Figure 2: Prey-Predators environment, same agents speeds. The upper plot shows predator agents performance comparison out of 10,000 sample episodes and 5 different models (2,000 sample episodes for each) while the bottom plot presents the Monte Carlo estimation of Shapley values (M=1,000) (mean over the 5 models) obtained for each predator agent (with "random" player exclusion method). Despite the agents having the same settings, a hierarchy in performance can still be discerned between them and this hierarchy is accurately reflected by Shapley values as seen in the upper plot, confirming RQ1.

3) Comparison of approximated Shapley Values with real Shapley values: In this subsection, we compare Monte-Carlo approximation of Shapley values with the *real* complete computation of Shapley values to verify RQ2. For that, we use Eq. 2 to compute Shapley values: marginal contributions are estimated as the mean global reward obtained by the coalition of players over a high number of episodes (i.e., 1,000 episodes for this experiment), with the same methods of action replacement as described in Section IV. Indeed, a large number of samples is needed for each coalition due to the stochastic nature of the environment, which leads to high variance in results. As depicted in Fig. 3, Shapley values estimated by Monte-Carlo sampling are very close to the real Shapley values with a difference of only 5% on average for all agents and models, while being simpler to implement and with a complexity that does not grow exponentially with the number of agents in the RL environment, thus supporting RQ2.

C. Experiment 2: Introducing Variations in Agents Speeds in Multiagent Particle

1) Environment Settings: In this experiment, we introduce variations in agent settings with the objective to disturb the actual distribution of contributions between agents obtained in experiment 1 and see if, as claimed in RQ1, this change will be reflected in the computed Shapley Values (i.e. we want to ensure that Shapley values correlate with the observed contributions). Thus, the speed of each predator will differ

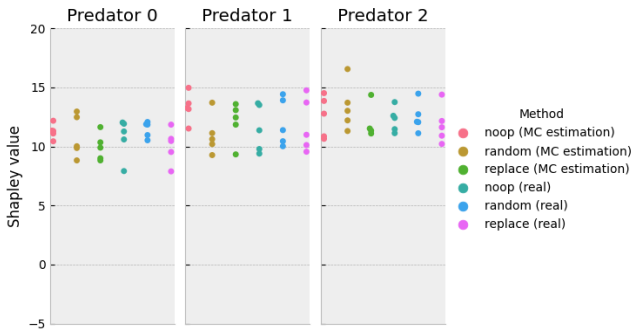


Figure 3: Prey-Predators environment, same agents speeds. Comparison of the Monte Carlo approximation of Shapley values with the real Shapley values. Each point relates to one of the 5 different models. Approximated values correlate with the real ones as hypothesized in RQ2.

from default ones so that we can expect that the faster agent will catch the prey more often and contribute the most to the global reward. In this w²manner, our goal is to create a clear hierarchy between agents so that correlating Shapley Values with the agents' observed behavior will be straightforward. We arbitrarily set the following speeds for each agent: Predator 0 (slow): 0.2, Predator 1 (medium): 0.8, Predator 2 (fast): 2.0. Statistics presented in Figure 4 show the performance of each predator agent in terms of the number of times each predator catches the prey. As expected, we can see that the faster agent (Predator 3) has a higher contribution than Predator 1, which has a higher contribution than Predator 0 (the slowest). More precisely, the ranking in speed is reflected in the ranking of contributions. Thus, this an ideal setting to test if RQ1 is valid. In addition, we also computed the real Shapley values for these settings in order to further verify RQ2.

2) Shapley values analysis: Figure 6 shows the Shapley values computed for each agent on each of the five models, over 1,000 sample episodes per model. This can be observed in a more convenient way in Figure 4, which shows the Shapley values computed for each agent on a single model. Following RQ1, the order of agents' Shapley Values is the following: Predator 2 then Predator 1 and finally Predator 0. These Shapley Values accurately correlates with the number of times each of the agents caught the prey (see Figure 4. In addition, as seen in Fig. 5, when making a single agent speed vary, we can observe that its Shapley value grows proportionally: the faster the agent is, the higher its Shapley value is. As expected, it makes sense, since it can catch the prey more easily, thus contributing more to the overall payout. Thus, here again, this experiment further supports RQ1, since Shapley values accurately corroborate the real (observed) distribution of contributions.

3) Comparison of approximated Shapley Values with real Shapley values: We conducted the same experiment described in Subsection V-B3 to further verify RQ2: we compared Monte-Carlo approximation of Shapley values with the *real* complete computation of Shapley values. Here again we can see in Fig. 6 that Shapley values estimated by Monte-Carlo sampling (with $M = 1,000$) are very close to the real ones with an average difference of 8% between the approximated values and the real

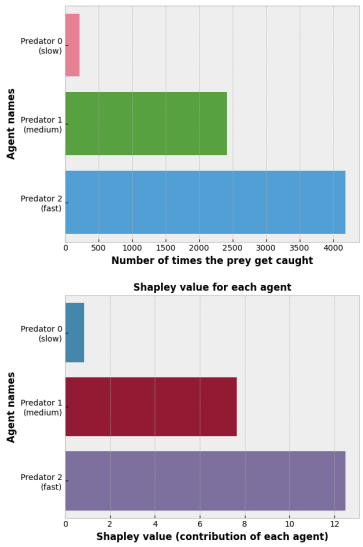


Figure 4: Prey-Predators environment, variable agents speeds. The upper plot shows predator agents performance comparison for 10,000 sample episodes and 5 different models (2,000 sample episodes for each of the five trained models) while the bottom plot presents the Monte Carlo estimation of Shapley values (M=1,000) for each predator agent (mean over the 5 models) with the *"random"* player exclusion option. Making the predators' speed vary created a clear hierarchy between them and, in the bottom plot, Shapley values accurately reflect this performance distribution supporting RQ1.

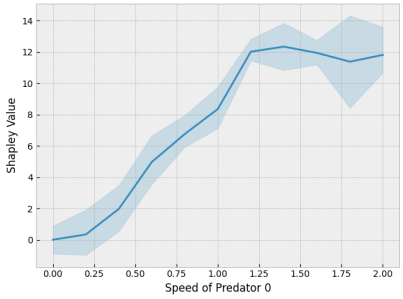


Figure 5: Prey-Predators environment, variable agents speeds. Monte Carlo approximation of Shapley values (M=1,000) obtained by a single MADDPG predator agent depending on its speed. It is clearly visible that the agent's Shapley value grows proportionally with its speed. As a faster predator can obviously catch more often the prey, this fact supports the claim that Shapley values correlate with agents contributions to the common goal, supporting RQ1.

ones for Predator 1 and Predator 2, when computed with the same replacement method. However, this percentage reaches 53 % when considering only Predator 0, because its Shapley values are very close to 0 with a small standard deviation, thus a small difference in value leads to a large difference percentage. Therefore, approximated Shapley values are close to the real ones while being simpler to implement and with a complexity that does not grow with the number of agents in the RL environment, supporting again RQ2.

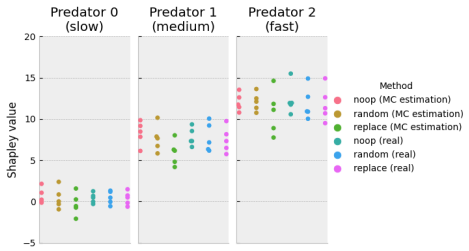


Figure 6: Prey-Predators environment, variable agents speeds. Comparison of the Monte Carlo approximation of Shapley values with the real Shapley values. Each point relates to one of the 5 different models. Approximated values correlate with the real ones as hypothesized in RQ2.

D. Experiment 3 (use case): Explaining Agent Contributions in Harvest environment

1) Environment Settings: In this experiment, we apply the Monte Carlo Shapley Value computation method to another multi-agent environment (i.e. *Harvest*). The goal here is to transpose this method in a use case where we try to explain and extract insightful information from a trained model in which all agents share the same settings. We leverage this data to attempt to further verify the research question described in Section I (RQ1).

We used the default settings of *Harvest*: 6 agents trying to collect apples on a pre-configured map. At first glance, this seems quite a large number of agents for a map this small (39*15, with 159 apples initially). Thus, we can hypothesize than some agents are superfluous, not contributing much to the global reward, and may even prevent other agents from elaborating effective strategies together, obstructing them in their movements.

2) Shapley values analysis: First, when looking at Figure 7, we can clearly see that Agent 5 does not seem to bring much added value to the team (Shapley value near to 0) while all the other agents seem to contribute a near equal amount to the global reward. In fact, while watching the agents playing, we can observe that the team found an efficient strategy for 4 agents but it could not find a use for the fifth one. Thus, he is left unaccounted for, wandering on the map, not harvesting any apple and sometimes randomly hitting the map border which grants him a negative reward. This may indicate that the default setting with 6 agents is erroneous: training only 5 agents could be enough to provide the same level of performance and will be less hardware and time consuming. Moreover, the fact that the first five agents contribute equally may be the proof that the *A3C* [1] training algorithm found a satisfying solution to distribute tasks between agents with the exception of Agent 5 who is left unaccounted for, maybe because none of its actions could help increase the global reward. Thus, we can conclude that, although limited, this synergy between agents, observed both in the obtained global reward and the near equal partition of Shapley values among agents, shows that agents are actually cooperating together. In fact, they are not being greedy by gathering too much apples at a time and killing trees as that would have prevented additional apples to spawn and thus would have stopped the growth of the global reward.

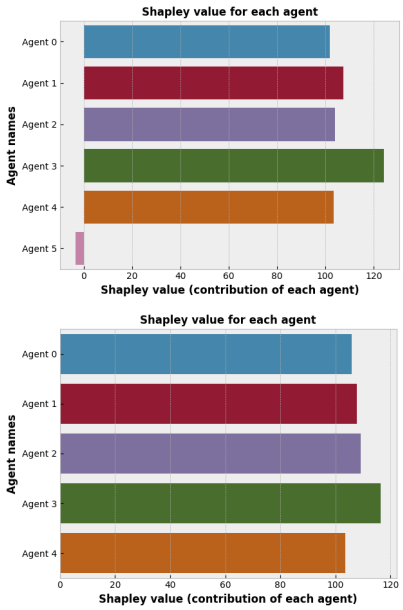


Figure 7: Harvest environment; The upper plot shows the Monte Carlo estimation of Shapley values (M=1,000) obtained for each *harvest* agent ("noop" action selection method) whereas the bottom plot also features Shapley values computed with the same settings but with Agent 5 deactivated. In the upper figure, Shapley values reveal that Agent 5 does not contribute at all and thus, might be superfluous in the team. That suspicion is verified in the bottom plot: the contribution distribution for Agents 0 to 4 remains very similar. Removing Agent 5 had very little incidence on the other agents and the global reward.

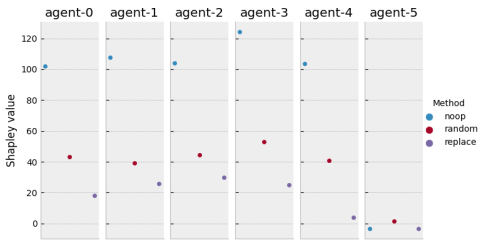


Figure 8: Harvest environment. Monte Carlo estimation of Shapley values (M=1,000) for each *harvest* agent using each of the three agent substitution methods. The same settings were used for all agents. *Noop* player exclusion method is the only one free from scoring negative cumulative reward due to random actions and thus is the one that would lead to Shapley values close to the agents' real contributions.

3) Following the insight given by Shapley Values: We saw that Shapley values analysis indicated that Agent 5 does not contribute at all to solve the common game (i.e. the *harvest* environment). So, we decided to re-run the Shapley values computation on the same model but this time with Agent 5 removed (i.e. completely deactivated and not appearing in the environment map) to see if, as suspected, the global reward and agents contributions remain the same. In Figures 7, we can see that Shapley values distribution when Agent 5 is deactivated stays nearly identical to the previous setting including it. Some very minor variation in Shapley values can be attributed to stochasticity, as every estimation of Shapley values does not yield the exact same results each time. In addition, the global reward remains stable too at around 450 (the mean global reward for all the sampled episodes is the sum of all the Shapley values, as explained in Subsection III-B). This means that removing Agent 5 did not have any negative effect on the game and corroborates our suspicion that its participation was not productive. Thus, we can conclude that we have been able to successfully derive a valuable insight from the analysis of Shapley values, strengthening the validity of RQ1.

4) Social outcome metrics: Analysis of the social behavior between agents: The goal here is to further explore RQ1 and see if agents practically leverage social learning and cooperate together. We present an analysis of how each social outcome metric (whose definitions are given in Subsection V-A) can be explained with our XAI approach, concretely, with Shapley values to explain multi-agent cooperative RL, with respect to the social metrics introduced in [18]. We excluded the Peace metric, also presented in [18], because it relies on a specific mechanism (i.e. *time-out* period during which a tagged agent cannot harvest apples anymore) that has been deemed of limited utility by the authors as the agents would learn very quickly not to use it. Note that the social outcome metrics (as originally defined in [18]) require credit assignment to be computed in a per-agent basis, which is not the case for Shapley values, which only supposes a global reward.

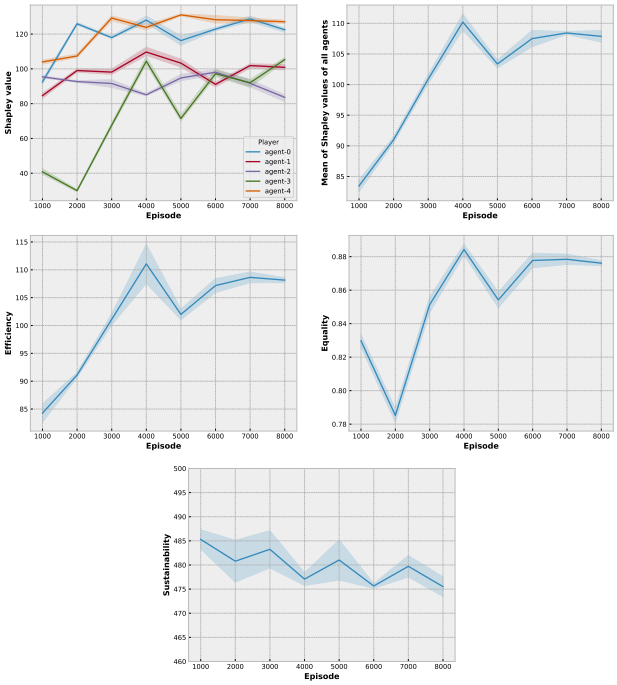


Figure 9: Harvest environment. Evolution of the Shapley values and social outcome metrics from [18] over different training episodes with A3C. From top to bottom are displayed, the Shapley values (*noop* action selection method; Monte Carlo estimation with M=500), the mean of those Shapley values of all agents, the efficiency metric, the equality metric and the sustainability metric.

For our experiments, we computed the three social outcome metrics presented above, run over 100 training episodes, as

well as Shapley values (using Monte-Carlo approximation and M=500), at different steps of the training (i.e. every 1,000 episodes), in order to analyze the evolution of those values during training. Results are presented in Figure 9. As the Efficiency metric is no more than the mean of the Shapley values (in expectation), we observe with no surprise that the mean of the Shapley values of all agents has the same evolution than the Efficiency, and directly correlates with it. The slight differences, such as the peak at episode 4,000, are due to the stochastic nature of the environment, the choice of M in Monte-Carlo approximation and the number of episodes used to compute the metrics.

We can conclude that the mean of Shapley values of all agents is a metric which explains the agent's efficiency and which does not assume any credit assignment between the agents: as stated in Equation 4, a shared global reward is enough. Equality evolves in the same manner (with a peak at episode 4,000 and a drop at episode 5,000) than the Efficiency of the agents and the mean of all Shapley values. This metric decreases a bit at episode 2,000: this fall is not captured by Efficiency and the mean of the Shapley Values. However, while looking at the Shapley values of each agent in Figure 9, we can clearly see that agent-3 obtains a lower reward than other agents. Using the Shapley values of all agents, we can thus explain that the decrease in the Equality at episode 2,000 is caused by the agent-3, which contributes little to the global reward (compared to other agents). We would have not been able to explain this behavior using Equality in a shared global reward context and herein the value of applying Shapley analysis in this context. As Shapley values are computed over the rewards of the team of agents, there is no link between Sustainability and Shapley values. In fact, the latter does not take in account the time at which the rewards are collected, as we can see in Equation 2.

We saw in this experiment that Shapley values can effectively capture both if the agents get high rewards (efficiency) and if the rewards are shared equally among all agents (equality). However, Shapley values cannot tell if the agents are obtaining the rewards continuously (Sustainability). Contrary to social outcome metrics, Shapley values can be computed even though the reward is globally shared between the agent. This is a huge advantage when the environment does not allow good credit assignment.

E. Choosing a player exclusion method

In this subsection, we analyze the results gathered about the player exclusion procedures (defined in Section IV) during the three experiments above, in order to answer RQ3 (i.e. select the method which seems the most appropriate overall). When looking at Figures 3, 6 and 8, we can see that the three player exclusion mechanisms lead to Shapley values that, when considered individually, are coherent from one agent to the other. However, in the referential of a single agent, the standard deviation between their respective values is important. In particular, there is a significant gap between *noop* and the other two methods (e.g. when focusing on Figure 8, an average gap of 73.1 reward units between *noop* and *random_player_action* while there is only an average gap of 20.5 reward units between *random* and *random_player_action*). This can be explained by the fact that randomly moving agents disturb the game way more than immobilized ones, as they can get negative rewards by hitting the map borders in *Multiagent Particle* or *killing* trees in *harvest* (i.e. harvesting all the apples that are contiguous). Thus, when using *random* or *replace*, a majority of coalitions are "parasited" by these negative rewards that contribute to lower the global reward and lead to lower Shapley values overall than the *noop* method (as observable in Figures 3, 6 and 8). So, in that context, *noop* action selection seems to be the most interesting method to get Shapley values assessing closely the agents' true contributions, free from negative reward noise (i.e. agents following the *noop* procedure never get negative rewards).

VI. Discussion

We described a Monte-Carlo approximation of Shapley value computation in cooperative games. The number of marginal contributions to estimate M, in order to compute the Shapley value for one player is a key element because its value will determine the algorithm complexity. A high M leads to an accurate estimation of Shapley values, while being slower to estimate. A low value of M results in a rougher, less precise Shapley value estimation (with a higher variance), but with a much faster computation time. Additionally, for a high enough value of M (e.g. 1,000 in our experiments) the approximated Shapley values are close enough to the real ones so that the ranking of agents contributions is preserved. Choosing a higher value (e.g. 2,000) would have increased computation time but also would have yielded Shapley values closer to the exact ones. Thus, the choice of the *optimal* M value may depend on the number of agents and the level of accuracy that is needed.

Furthermore, we demonstrated the usefulness of Shapley values for explaining RL models in Section V. In fact, they provide a form of explanations (i.e. discrete values) that are very understandable by researchers and developers as they represent portions of the reward value of the agents team, partitioned according to each agent contribution. They could also provide explanations to the general public (e.g. human users of a cooperative RL-based system) that could perceive them as the intrinsic "value" of each agent and reassure them on the effectiveness of the system. Moreover, Shapley values could be a good way to detect biases in the training of an RL model, since they require to analyze the individual behavior of each agent and thus could highlight disparities between their different strategies.

Concerning the player exclusion method to replace missing agents from a coalition, *noop* (no-operation) action seems to be the most neutral, interaction-free method when the environment offer this possibility, as random methods are prone to get high negative rewards and interfere in the game. We also investigated social interactions between agents and found that Shapley values are able to effectively capture both Efficiency and Equality metrics, while still being able to be computed even though the reward is globally shared between agents. This is a huge advantage when the environment does not enable

fair individual-level credit assignment. So, in consequence, we can assert that Shapley Values are an effective way to explain the contributions of RL agents, and, to some extent, the relationships between them.

However, our approach is limited to multi-agent cooperative RL and in its current form cannot be applied to competitive and single-agent models. In addition, it cannot be used to explain an agent's actions nor explain a specific episode of interest, as it only provides an average metric for the contribution of each agent in a cooperative game, with the total of Shapley values corresponding to the mean global reward of the grand coalition (i.e. the one containing all agents). Thus, it must be be considered as a way to get a contribution index of agents of a model. Finally, while the method we implemented to estimate Shapley values using Monte-Carlo approximation is more effective than computing the exact Shapley values, it still remains time consuming and it would be interesting to focus more on performances in a future work (see Table II).

VII. Conclusion and Future Work

We tested and confirmed all of our 3 hypotheses (presented in Section I), conducted in two socially challenging multi-agent RL environments (*Harvest* from Sequential Social Dilemmas [17], [52] and *Particle Multiagent* [16]) and two different RL algorithms (i.e. MADDPG [16] and A3C [1]). We showed that Shapley values computation could be a potential breakthrough elucidating understanding in multiagent XRL environments, as it can efficiently assess the contribution of agents to the global reward in cooperative settings, and also provides insightful information about the agents behaviours and their social interactions. As a promising lead, hybrids of explainability and RL methods have been proposed as in [26]. However, they use Shapley values for RL in a totally different approach where they help create a novel Q-function to derive a new variant of DDPG with a focus set not on explainability or interpretability but performance (see Section II for additional details).

Nonetheless, numerous issues remain to explore in future work. Different interpretations of Shapley values to explain deep RL must be proposed to increase the levels of explanation granularity. Robustness remains a critical issue for XRL (and XAI in a more general sense) and other statistical methods could prove very useful for that purpose, as presented in [54]–[56] (e.g. Winsorised or trimmed estimators). Moreover, Shapley values could also be combined with a robust model selection measure such as the Lorenz Zonoids as presented in [23], [57]. Besides, Shapley values or other additive and non additive methods could be used to explain the roles taken by agents when learning a policy to achieve a collaborative task, but also to detect defects in agents while training, or the fed data, as highlighted in Section V-D. Furthermore, we could take into account the dynamic nature of RL (vs. the static settings of most ML models where only a single data point needs to be explained) and create a novel approach that evaluates the contributions of agents through time (during evaluation for example). Here, "temporal" Shapley values could be approximated with a model as in [26]. However, we would lose one of the main advantages of a post-hoc XAI method (i.e. being agnostic to the RL algorithm), as the Shapley prediction model would be specific to the RL model used. Finally, we could also explore a different contribution ranking scheme than the one presented in Section IV (order of importance in the team). For instance, we could rank each set of observation experiences per agent in order of average importance (by making coalitions of agent observation trajectories or strategies to estimate marginal contributions on the mean final reward).

We demonstrated that Shapley values based attribution is a versatile ingredient to include towards the future development of post-hoc XAI techniques, but other techniques for different targeted audience (developer, tester, end-user, general public) need yet to be developed. Only in this way will we produce actionable explanations and more comprehensive frameworks for explainable RL that are deployable in real life problems, that could include critical decision making (in crisis contexts such as pandemics [15] for instance) or policy making under fairness/nondiscriminatory constraints [58].

References

[1] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," 2016. [Online]. Available: https://arxiv.org/pdf/1602.01783

[2] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, S. Legg, and K. Kavukcuoglu, "Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures," 2018. [Online]. Available: https://arxiv.org/pdf/1802.01561

[3] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. R. Baker, M. Lai, A. Bolton, Y. chen, T. P. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, pp. 354–359, 2017. [Online]. Available: https://www.nature.com/articles/nature24270

[4] E. Puiutta and E. M. Veith, "Explainable reinforcement learning: A survey," 2020.

[5] A. Heuillet, F. Couthouis, and N. Díaz-Rodríguez, "Explainability in deep reinforcement learning," *Knowledge-Based Systems*, vol. 214, p. 106685, 2021. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705120308145

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," 2016.

[7] S. El-Sappagh, J. M. Alonso, S. R. Islam, A. M. Sultan, and K. S. Kwak, "A multilayer multimodal detection and prediction model based on explainable artificial intelligence for alzheimer's disease," *Scientific reports*, vol. 11, no. 1, pp. 1–26, 2021.

[8] H. Arnout, M. El-Assady, D. Oelke, and D. A. Keim, "Towards a rigorous evaluation of xai methods on time series," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 2019, pp. 4197–4201.

[9] Z. Juozapaitis, A. Koul, A. Fern, M. Erwig, and F. Doshi-Velez, "Explainable reinforcement learning via reward decomposition." [Online]. Available: http://web.engr.oregonstate.edu/~afern/papers/reward_decomposition__workshop_final.pdf

[10] S. Greydanus, A. Koul, J. Dodge, and A. Fern, "Visualizing and understanding atari agents," 2017. [Online]. Available: https://arxiv.org/pdf/1711.00138

[11] P. Madumal, T. Miller, L. Sonenberg, and F. Vetere, "Explainable reinforcement learning through a causal lens," 2019. [Online]. Available: https://arxiv.org/pdf/1905.10958.pdf

[12] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017. [Online]. Available: https://arxiv.org/pdf/1705.07874

[13] L. S. Shapley, "The value of an n-person game," 1951. [Online]. Available: https://www.rand.org/content/dam/rand/pubs/research_memoranda/2008/RM670.pdf

[14] G. Hardin, "The tragedy of the commons," *Journal of Natural Resources Policy Research*, vol. 1, no. 3, pp. 243–253, 2009.

[15] M. Chica, J. M. Hernández, and J. Bulchand-Gidumal, "A collective risk dilemma for tourism restrictions under the covid-19 context," *Scientific Reports*, vol. 11, no. 1, pp. 1–12, 2021.

[16] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Neural Information Processing Systems (NIPS)*, 2017. [Online]. Available: https://arxiv.org/pdf/1706.02275.pdf

[17] J. Z. Leibo, V. Zambaldi, M. Lanctot, J. Marecki, and T. Graepel, "Multi-agent reinforcement learning in sequential social dilemmas," 2017.

[18] J. Perolat, J. Z. Leibo, V. Zambaldi, C. Beattie, K. Tuyls, and T. Graepel, "A multi-agent reinforcement learning model of common-pool resource appropriation," 2017.

[19] E. Hughes, J. Z. Leibo, M. G. Phillips, K. Tuyls, E. A. Duéñez-Guzmán, A. G. Castañeda, I. Dunning, T. Zhu, K. R. McKee, R. Koster *et al.*, "Inequity aversion improves cooperation in intertemporal social dilemmas," *arXiv preprint arXiv:1803.08884*, 2018.

[20] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, and F. Giannotti, "A survey of methods for explaining black box models," 2018.

[21] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," 2019. [Online]. Available: https://arxiv.org/pdf/1910.10045.pdf

[22] O.-M. Camburu, E. Giunchiglia, J. Foerster, T. Lukasiewicz, and P. Blunsom, "The struggles of feature-based explanations: Shapley values vs. minimal sufficient subsets," 2020.

[23] P. Giudici and E. Raffinetti, "Shapley-lorenz decompositions in explainable artificial intelligence," *SSRN Electronic Journal*, 01 2020.

[24] Z. Mai, D. Shim, J. Jeong, S. Sanner, H. Kim, and J. Jang, "Adversarial shapley value experience replay for task-free continual learning," 2020.

[25] M. Sundararajan and A. Najmi, "The many shapley values for model explanation," 2019. [Online]. Available: https://arxiv.org/abs/1908.08474

[26] J. Wang, Y. Zhang, T.-K. Kim, and Y. Gu, "Shapley q-value: A local reward approach to solve global reward games," 2019. [Online]. Available: https://arxiv.org/pdf/1907.05707.pdf

[27] A. Harutyunyan, W. Dabney, T. Mesnard, M. Azar, B. Piot, N. Heess, H. van Hasselt, G. Wayne, S. Singh, D. Precup *et al.*, "Hindsight credit assignment," *arXiv preprint arXiv:1912.02503*, 2019.

[28] M. Staniak and P. Biecek, "Explanations of model predictions with live and breakdown packages," *arXiv preprint:1804.01955*, 2018. [Online]. Available: https://arxiv.org/pdf/1804.01955.pdf

[29] A. Gosiewska and P. Biecek, "Do not trust additive explanations," *arXiv preprint arXiv:1903.11420*, 2019.

[30] A. Haydari and Y. Yilmaz, "Deep reinforcement learning for intelligent transportation systems: A survey," 2020.

[31] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," 2021.

[32] H. Nguyen and H. La, "Review of deep reinforcement learning for robot manipulation," 02 2019, pp. 590–595.

[33] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. D. Rodríguez, "Continual learning for robotics," *ArXiv*, vol. abs/1907.00182, 2019. [Online]. Available: https://arxiv.org/pdf/1907.00182.pdf

[34] J. Duffy and J. Ochs, "Cooperative behavior and the frequency of social interaction," *Games and Economic Behavior*, vol. 66, no. 2, pp. 785 – 812, 2009, special Section In Honor of David Gale. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0899825608001395

[35] A. M. Colman, "ooperation, psychological game theory, and limitations of rationality in social interaction," *The Behavioral and brain sciences*, vol. 26(2), p. 139–198, 2003.

[36] N. Jaques, A. Lazaridou, E. Hughes, Ç. Gülçehre, P. A. Ortega, D. Strouse, J. Z. Leibo, and N. de Freitas, "Intrinsic social motivation via causal influence in multi-agent RL," *CoRR*, vol. abs/1810.08647, 2018. [Online]. Available: http://arxiv.org/abs/1810.08647

[37] K. Ndousse, D. Eck, S. Levine, and N. Jaques, "Learning social learning," 2020. [Online]. Available: https://arxiv.org/abs/2010.00581

[38] C. Molnar, *Interpretable Machine Learning*, 2019, https://christophm.github.io/interpretable-ml-book/.

[39] *The Shapley Value: Essays in Honor of Lloyd S. Shapley*. Cambridge University Press, 1988. [Online]. Available: http://www.library.fa.ru/files/Roth2.pdf

[40] E. Friedman and H. Moulin, "Three methods to share joint costs or surplus," *Journal of Economic Theory*, vol. 87, no. 2, pp. 275 – 312, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022053199925346

[41] N. Bard, J. N. Foerster, S. Chandar, N. Burch, M. Lanctot, H. F. Song, E. Parisotto, V. Dumoulin, S. Moitra, E. Hughes *et al.*, "The hanabi challenge: A new frontier for ai research," *Artificial Intelligence*, vol. 280, p. 103216, 2020.

[42] J. K. Terry, B. Black, M. Jayakumar, A. Hari, R. Sullivan, L. Santos, C. Dieffendahl, N. L. Williams, Y. Lokesh, C. Horsch *et al.*, "Pettingzoo: Gym for multi-agent reinforcement learning," *arXiv preprint arXiv:2009.14471*, 2020.

[43] J. Suarez, Y. Du, I. Mordach, and P. Isola, "Neural mmo v1. 3: A massively multiagent game environment for training and evaluating neural networks," *arXiv preprint arXiv:2001.12004*, 2020.

[44] T. Chu, S. Qu, and J. Wang, "Large-scale multi-agent reinforcement learning using image-based state representation," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 7592–7597.

[45] U. Faigle and W. Kern, *The Shapley value for cooperative games under precedence constraints*, ser. Memorandum. University of Twente, Faculty of Mathematical Sciences, 1992, no. 1025.

[46] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowledge and information systems*, vol. 41, no. 3, pp. 647–665, 2014. [Online]. Available: https://moodle.telekom.ftn.uns.ac.rs/pluginfile.php/13342/mod_folder/content/0/Feature%20importance%20paper.pdf?forcedownload=1

[47] A. Tallón-Ballesteros and C. Chen, "Explainable ai: Using shapley value to explain complex anomaly detection ml-based systems," *Machine Learning and Artificial Intelligence: Proceedings of MLIS 2020*, vol. 332, p. 152, 2020.

[48] I. E. Kumar, S. Venkatasubramanian, C. Scheidegger, and S. Friedler, "Problems with shapley-value-based explanations as feature importance measures," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5491–5500.

[49] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015. [Online]. Available: https://arxiv.org/abs/1509.02971

[50] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," 2017.

[51] C. S. de Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. Torr, M. Sun, and S. Whiteson, "Is independent learning all you need in the starcraft multi-agent challenge?" *arXiv preprint arXiv:2011.09533*, 2020.

[52] E. Vinitsky, N. Jaques, J. Leibo, A. Castenada, and E. Hughes, "An open source implementation of sequential social dilemma games," https://github.com/eugenevinitsky/sequential_social_dilemma_games/issues/182, 2019, gitHub repository.

[53] C. Gini, "Variabilità e mutabilità," *Reprinted in Memorie di metodologica statistica (Ed. Pizetti E*, 1912.

[54] P. J. Huber, *Robust statistics*. John Wiley & Sons, 2004, vol. 523.

[55] R. A. Maronna, R. D. Martin, V. J. Yohai, and M. Salibián-Barrera, *Robust statistics: theory and methods (with R)*. John Wiley & Sons, 2019.

[56] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust statistics: the approach based on influence functions*. John Wiley & Sons, 2011, vol. 196.

[57] P. Giudici and E. Raffinetti, "Shapley-lorenz explainable artificial intelligence," *Expert Systems with Applications*, vol. 167, p. 114104, 2021.

[58] N. Díaz-Rodríguez, R. Binkytė-Sadauskienė, W. Bakkali, S. Bookseller, P. Tubaro, A. Bacevicius, and R. Chatila, "Questioning causality on sex, gender and covid-19, and identifying bias in large-scale data-driven analyses: the bias priority recommendations and bias catalog for pandemics," 2021.

[59] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel, "Value-decomposition networks for cooperative multi-agent learning," 2017.

ACKNOWLEDGEMENTS

We would like to thank Frédéric Herbreteau, Adrien Bennetot and Léo Heidelberger for their help and support. We also thank reviewers for their valuable comments and suggestions.

APPENDIX

ADDITIONAL RESULTS

We conducted extra experiments attempting to test the scalability of models in Experiment 2 (see Subsection V-C) by

increasing the number of agents to 9 predators and 3 preys. However, running over 500,000 episodes MADDPG did not make the predators global reward converge. When observing the model, contrary to the one used in Experiment 1 (see Subsection V-B) agents do not show any collective cooperative strategy and wander aimlessly on the map failing to catch the prey (except when randomly hitting them). We hypothesize that the high number of agents makes cooperation between them difficult and this may be pointed out as a limitation of MADDPG.

We also tried to use Value-Decomposition Networks (VDN) [59] to train a model for the Harvest [52] environment but this was not successful since the model struggled to converge and did not converged to a positive mean reward while A3C reached around 500 of mean reward, see Fig. 15).

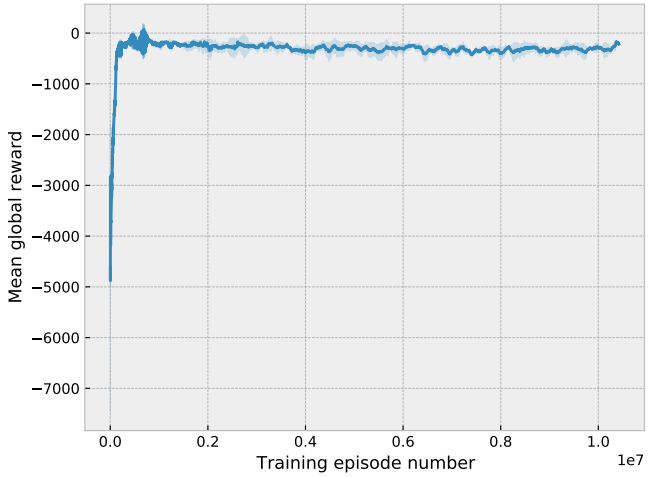


Figure 10: Learning curve of the Harvest VDN model (averaged over 3 runs). We can clearly see that the model did not reach a way lower max reward than the A3C model in Fig. 15

Social Metrics Per Agent

We further studied the social metrics presented in Subsection V-D4 and introduced in [18]. We refactored them into a per agent basis in order to make a more relevant comparison with Shapley values. Considering agent i among N agents, we obtained the following formulas:

$$U_i = \mathbb{E}\left[\frac{R_i}{T}\right] \tag{8}$$

$$S_i = t_i = \mathbb{E}[t/r_t^i > 0] \tag{9}$$

$$E_i = 1 - \frac{\sum_{j=1}^{N} |R_i - R_j|}{2\sum_{j=1}^{N} R_j} \tag{10}$$

Where R_i is the reward obtained by agent i. Using these refactored formulas, we plotted the metrics using the same data than in Subsection V-D4 in Fig. 11, Fig. 12 and Fig. 13.

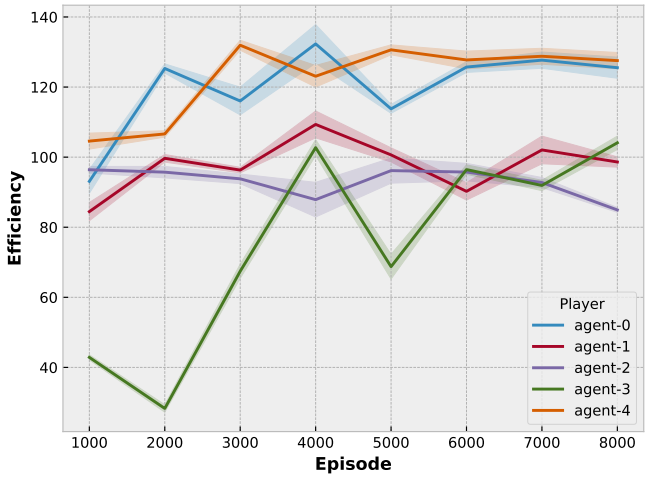


Figure 11: Evolution of *Efficiency* social metric per agent over several episodes (averaged on 4 runs).

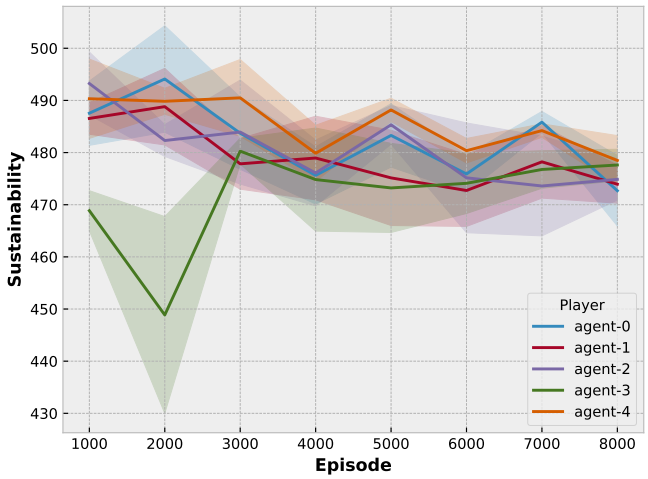


Figure 12: Evolution of Sustainability per agent over several episodes (averaged on 4 runs).

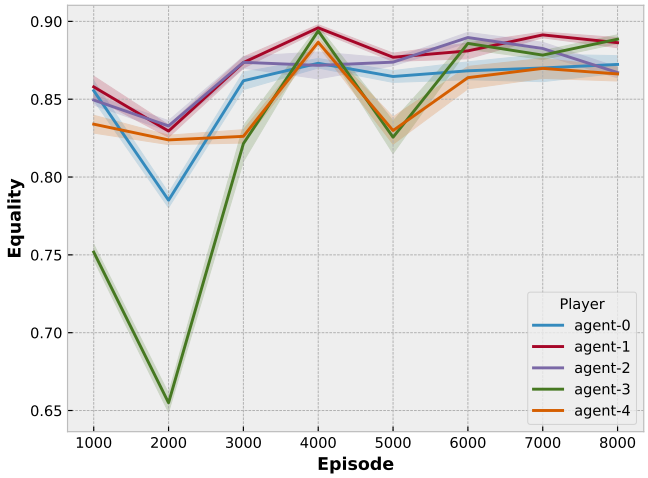


Figure 13: Evolution of Equality per agent over several episodes (averaged on 4 runs).

When analyzing these figures, we can clearly see that

Efficiency (Fig. 11) is identical to the plot of Shapley values (Fig. 9). That is a logical result since the efficiency per agent (Eq. 8) computes the same result than Shapley values: the average contribution (reward) per agent. Equality (Fig. 13) also follows the same trend as Shapley values. This visual correlation supports our claim that Shapley values is a relevant tool to assert the contribution of agents in RL cooperative settings.

Experimental Details

Here we present some additional details about the setup of the experiments showcased in Section V.

Parameter	Value
Learning Rate	0.01
Optimizer	Adam
Number of MLP units	128
Discount Factor	0.95
Batch Size	1024

Table III: Hyperparameters used for every MADDPG and DDPG model on Predator-Prey scenario. These are the default parameters recommended by [16]. Other hyperparameters (e.g., the number of predators or their speed) may vary and their values are indicated in the experimental settings description (Subsections V-B1 and V-C1).

Parameter	Value
Learning Rate	0.0001
Optimizer	Adam
Number of MLP units	128
Discount Factor	0.99
Batch Size	30000

Table IV: Hyperparameters used for A3C models on Harvest scenario. These are the default parameters recommended by [52]. Other hyperparameters (e.g. the number of agents) may vary and their values are indicated in the experimental settings description (see Subsection V-D1).

Figure 14: Learning curve of the Prey-Predator MADDPG models (average over 5 models) used in Experiment 1.

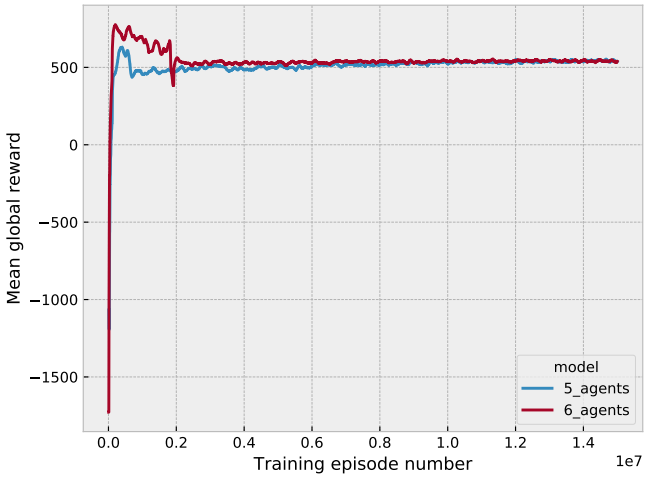


Figure 15: Learning curve of the Harvest A3C models (5 and 6 agents models used in Experiment 3). We can clearly observe that both models quickly converge to the same reward.