On Diagnostics for Understanding Agent Training Behaviour in Cooperative MARL

Wiem Khlifi[1, 2], **Siddarth Singh**[1], **Omayma Mahjoub**[1], **Ruan de Kock**[1] **Abidine Vall**[3] **Rihab Gorsane**[1] **Arnu Pretorius**[1]

[1]InstaDeep Ltd
[2]National School of Computer Science, Tunisia
[3]National School of Engineering of Tunis

Abstract

Cooperative multi-agent reinforcement learning (MARL) has made substantial strides in addressing the distributed decision-making challenges. However, as multi-agent systems grow in complexity, gaining a comprehensive understanding of their behaviour becomes increasingly challenging. Conventionally, tracking team rewards over time has served as a pragmatic measure to gauge the effectiveness of agents in learning optimal policies. However, we argue that relying solely on the empirical returns may obscure crucial insights into agent behaviour. In this paper, we explore the application of explainable AI (XAI) tools to gain profound insights into agent behaviour. We employ these diagnostics tools within the context of Level-Based Foraging and Multi-Robot Warehouse environments and apply them to a diverse array of MARL algorithms. We demonstrate how our diagnostics can enhance the interpretability and explainability of MARL systems, providing a better understanding of agent behaviour.

Introduction

Multi-agent Reinforcement Learning (MARL) has shown immense promise in tackling complex decision-making challenges across various domains, such as robotics, healthcare, and energy. For example, MARL has been used to coordinate robotic surgery teams (Scheikl et al. 2021), optimise smart grid operations (Roesch et al. 2020), and manage air traffic flow (Vidhate and Kulkarni 2017).

Standard methods for evaluating MARL in the cooperative setting rely solely on measuring empirical returns or statistics derived from them (Gorsane et al. 2022). Although tracking returns over time is a reasonable way to measure whether agents are learning optimal policies, recent scrutiny has revealed the limitations of evaluation methods, prompting the development of more robust solutions (Whiteson et al. 2011; Agarwal et al. 2022; Gorsane et al. 2022). Yet these standard measures often fail to explain the underlying behaviour of MARL algorithms (Canese et al. 2021; Son et al. 2022). Addressing this challenge is where the growing field of Explainable Artificial Intelligence (XAI) steps in.

XAI consists of machine learning (ML) techniques that can provide human interpretable insights into the inner workings of ML models (Arrieta et al. 2020). It has been extensively explored in single-agent RL (Heuillet, Couthouis, and Díaz-Rodríguez 2021; Hu et al. 2022; Vouros 2022; Puiutta and Veith 2020; Dazeley, Vamplew, and Cruz 2023), where various methods have been proposed to analyse and visualise the policies, value functions, and salient features of individual agents. However, explainability remains underrepresented in the context of MARL, where agents interact with each other and the environment in complex and dynamic ways. Existing tools for MARL explainability are limited and specialised for specific scenarios or algorithms (Shapley 1953; Heuillet, Couthouis, and Díaz-Rodríguez 2022). Therefore, there is a need for more general and robust tools that can enhance interpretability and understanding of MARL systems.

In this paper, we argue that it is necessary not to rely on empirical performance for the evaluation of MARL algorithms; instead, we can supplement performance analysis with XAI tools to provide direct insights into agent behaviour to provide a more holistic view of results. To accomplish this, we adapt multiple simple implementation-agnostic methods for analyzing behaviour and we demonstrate how they can detect and offer valuable insights. Specifically, we assess learning stability using Policy Entropy and Agent Update Divergence and analyse a Task Switching metric to provide insights into the learned action distributions of MARL agents. From this analysis in the Level-Based Foraging (LBF) and Multi-Robot Warehouse(RWARE) domains (Papoudakis et al. 2021), we uncover insights into how premature convergence limits the performance of Multi-Agent Proximal Policy Optimisation (MAPPO) on LBF compared to Multi-Agent Advantage Actor-Critic (MAA2C) and show how the limitations of Q-learning methods on RWARE are not purely a byproduct of sparsity as originally hypothesised by (Papoudakis et al. 2021).

Diagnostics Tools

To assess agents' behaviour during training or evaluation, we investigate a diverse set of tools used as diagnostics of policy learning. We provide an overview of these tools which are typically based on metrics derived from prior work that we adapt to the MARL setting.

Policy Entropy. (Abdallah 2009) demonstrated that only tracking global returns may mask potential policy instability issues in agents. To address this, the authors proposed using Policy Entropy as a measure of stability during learning. This metric is used to measure the uncertainty or randomness of a stochastic policy of an agent i, which is a probability distribution over actions given an observation and we can compute the Policy Entropy of agent i's policy π_i as follows

$$H(\pi_i) = -\sum_{a_i \in A_i} \pi_i(a_i|o_i) \log \pi_i(a_i|o_i), \qquad (1)$$

where o_i is the agent's observation and $a_i \in A_i$ denotes an action from the agent's action set A_i.

Intuitively, this measures how spread out the probabilities, given by a policy π_i, are across different actions. If the probabilities are evenly distributed (i.e., the policy is highly uncertain), the entropy is higher and it indicates that the agent's choices are more diverse and exploratory, as it is not favouring any specific action strongly. Contrarily, if the probabilities are concentrated on a few actions (i.e., the policy is more deterministic), the entropy is lower and it indicates that the agent tends to concentrate its probability on a smaller subset of actions.

Agent Update Divergence. The per-step KL divergence is used as a metric to assess the policy changes of each agent throughout the training process. It provides insights into the stochastic nature of the policy, serving as a monitor of the evolution of the policy. A persistent high KL divergence observed over time indicates a propensity for a stochastic policy that persists in exploring varied strategies. Conversely, a low KL divergence implies that the agent's policy is trending towards predictability, converging either towards a deterministic behavioural or, possibly, a suboptimal strategy. If we define the cross-entropy between two policies π and π' as $H(\pi, \pi') = -\sum_{a \in A} \pi(a|o) \log \pi'(a|o))$, the update divergence for each agent i is given by

$$D_{kl}(\pi_i \| \pi'_i) = H(\pi_i, \pi'_i) - H(\pi_i), \qquad (2)$$

where π_i and π'_i are the current and old policy of agent i.

Task Switching. During training agents often assume distinct roles (Wang et al. 2020; Phan et al. 2021). While some MARL methods such as ROMA (Wang et al. 2020) explicitly account for this, it is not a typical feature of algorithms. Although the above KL divergence has been proposed as a measure of agent action diversity, it can vary even among agents with similar policies due to observation differences (Hu et al. 2022). For example, in a football game, defenders with identical roles may respond differently based on unique observations (Kurach et al. 2020). Therefore, we explore implementation-agnostic methods to monitor agent diversity in MARL training by using a Task Switching tool. We implement such a tool by tracking the usage of each agent's action at each timestep t during the evaluation phase and computing the overall likelihood of each agent's action being selected as

$$\mathcal{T}_i = \sigma\left(\sum_{t=0}^{T} \mathbf{a}_i^t\right), \qquad (3)$$

where $\mathbf{a}_i^t$ represents the one-hot encoded vector of the action chosen by agent i at timestep t with T being the terminal timestep of the evaluation phase and σ the softmax function.

A Task Switching metric makes it possible to quantify the frequency at which an agent selects specific actions during the evaluation phase and it can be useful in diagnosing the learning of agents by measuring how often they use the different actions.

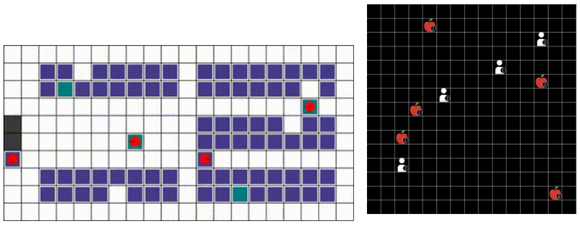


Figure 1: **Top**: Multi-Robot Warehouse (RWARE). **Bottom**: Level-based foraging (LBF)

The Task Switching tool is complementary to the Policy Entropy and Agent Update Divergence tools, since it measures the actual behaviour of the agents after acquiring some knowledge, during the evaluation phase, rather than during the learning phase in addition to the fact that the actual behaviour of the agents depends not only on their policies, but also on their observations and roles in the environment.

Experiment

We conducted experiments using the same training setup as (Papoudakis et al. 2021). We used the optimized hyperparameters and the EPyMARL framework, following the evaluation protocol proposed by (Gorsane et al. 2022). In addition, we trained a range of algorithms, from Q-learning to policy gradient (PG) methods, across seven distinct scenarios in the Level-Based Foraging (LBF) environment and three distinct scenarios in the Multi-Robot Warehouse (RWARE), using 10 different seeds. Experiment results are depicted in Figure 2.

Environment. In our study, we directed our attention toward the two environments, LBF and RWARE, as outlined in prior works (Albrecht and Ramamoorthy 2015; Albrecht and Stone 2019; Papoudakis et al. 2021). In these environments, the reward signal is notably sparse, rendering it a more challenging task during the learning phase compared to environments like the Starcraft Multi-Agent Challenge (SMAC) (Samvelyan et al. 2019; Papoudakis et al. 2021).

LBF represents a mixed cooperative-competitive environment that emphasises coordination between agents. As illustrated in Figure 1, agents are placed within a grid world and assigned different levels. To collect food, the cumulative level of participating agents must meet or exceed the food's designated level. Agents receive points equivalent to the level of the collected food and their own level.

RWARE (Christianos, Schäfer, and Albrecht 2020; Papoudakis et al. 2021) is a multi-agent environment that is designed to represent a simplified setting where robots move goods around a warehouse. This environment requires agents (circles) to move requested shelves (green squares) to the goal (dark squares) and back to an empty square as illustrated at the top of Figure 1. Tasks are partially observable with a very sparse reward signal as agents have limited sight and are rewarded only upon successful delivery.

Algorithms. Our analysis focuses on a subset of commonly used MARL algorithms, including both centralized training with decentralized execution (CTDE) and independent learning (IL) methods (Zhang and Yu 2020; Gronauer and Diepold 2021; Annie et al.). These algorithms belong to two categories: Q-learning-based algorithms, such as Independent Q-Learning (IQL) (Tan 1997), Value-Decomposition Networks (VDN) (Sunehag et al. 2017), and QMIX (Rashid et al. 2018), and PG algorithms, such as Multi-Agent Proximal Policy Optimization (MAPPO) (Yu et al. 2022) and Multi-Agent Advantage Actor-Critic (MAA2C) (Foerster et al. 2018). We employed *parameter sharing* across all algorithms during training which was shown by (Papoudakis et al. 2021) to lead to greater episode returns when compared to cases without parameter sharing.

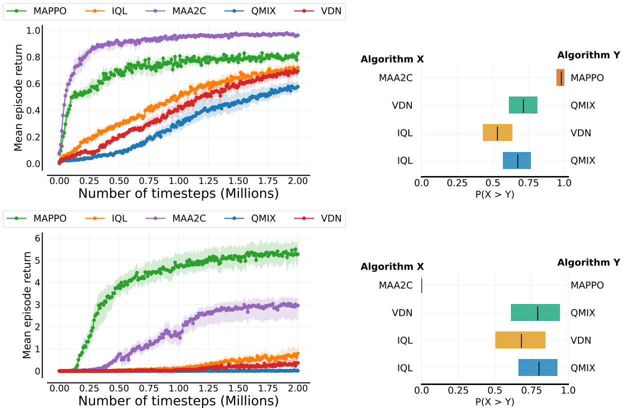
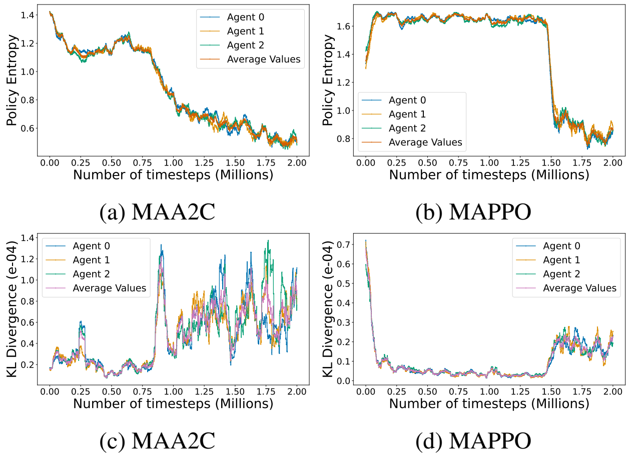


Figure 2: *Algorithm performance on LBF and RWARE including probability of improvement(right), and sample efficiency curves(left).* **Top row:** Performance of algorithms on 7 LBF tasks. **Bottom row:** Performance of all algorithms on 3 RWARE tasks.

Case Studies

After running experiments on seven scenarios in the LBF environment, each with ten different seeds, our results point out that relying solely on empirical returns may not offer a complete understanding of algorithmic performance in MARL. However, when complemented by simple XAI methods, these results can provide valuable insight into the environmental and algorithmic factors of the MARL setting. In the following, from Figure 3 to 5, different agents have similar behaviour which indicates that they have learned similar or redundant policies that do not vary much across scenarios and this outcome can be attributed to the use of parameter sharing in the learning process.

Why does MAA2C outperform MAPPO? We attempt to answer this question using the above diagnostic. We focus on the LBF scenario *Foraging-15x15-3p-5f*, using one seed, to examine the performance disparities between MAA2C and MAPPO. Based on the fact that the core architecture of PPO (Yu et al. 2022) extends A2C by introducing a clipping function, it is plausible to attribute the reduced performance of MAPPO to its sensitivity to the clipping ratio, which can lead to an early suboptimal policy. A smaller clipping ratio can make PPO overly cautious, slowing down the learning process. Conversely, a larger clipping ratio can make PPO overly aggressive, potentially destabilising the learning process. This hypothesis is supported by the slower convergence of the MAPPO algorithm across all LBF scenarios, which initially struggles to improve its policy, as shown in Figure 3 and 4. Furthermore, MAPPO continues to face challenges in reaching a performance level comparable to MAA2C in the LBF environment, as shown in Figure 2, where the probability that MAA2C outperforms MAPPO in all LBF scenarios on 10 different seeds is close to 1.



(a) MAA2C

(b) MAPPO

(c) MAA2C

(d) MAPPO

Figure 3: *Training stability results on Foraging-15x15-3p-5f (one seed).* **Top row:** Policy Entropy. **Bottom row:** Agent Update Divergence.

The learning dynamics, characterised by the Agent Update Divergence and Policy Entropy metrics, exhibit notable differences between agents trained under the MAA2C and MAPPO algorithms. Specifically, MAA2C demonstrates a more effective learning trajectory, marked by consistent steps in policy improvement.

In Figures 3a and 3c, as well as in the average values across all agents, we observe distinct patterns for agents trained with MAA2C. Initially, their learning progresses slowly, but subsequently, their policies undergo significant changes. This phenomenon can be attributed to continuous exploration of the environment, leading to the discovery of more effective strategies. Moreover, the increased certainty in action selection suggests successful policy updates. These metrics collectively indicate that the policies remain stochastic and exploratory, evolving frequently over time without full convergence.

In contrast, when examining the behaviour of agents trained with the MAPPO algorithm in Figures 3b and 3d, we observe a different pattern. It appears that the sensitivity to the clipping ratio in MAPPO may hinder the agents from converging rapidly, in addition to the fact that the environment is highly random (levels and coordinates). The policy may become trapped in a local optimum early in training, persisting for over one million steps. This situation limits the algorithm's ability to explore effectively and discover a superior policy during the initial stages of training.

In Figure 4, action 0 corresponds to the "no operation" action, action 5 is used to pick up nearby food, and the remaining actions are movement actions. The *Task Switching* plots reveal that MAA2C becomes more sensitive to changes in action selection, as indicated by the converging action probabilities. This enhanced sensitivity allows MAA2C to explore and utilise a broader range of actions, leading to a reduction in actions that are associated with lower rewards. In contrast, MAPPO exhibits greater action dominance, particularly in one action (action 4). The differences in performance between MAA2C and MAPPO can be attributed to their distinct learning strategies.

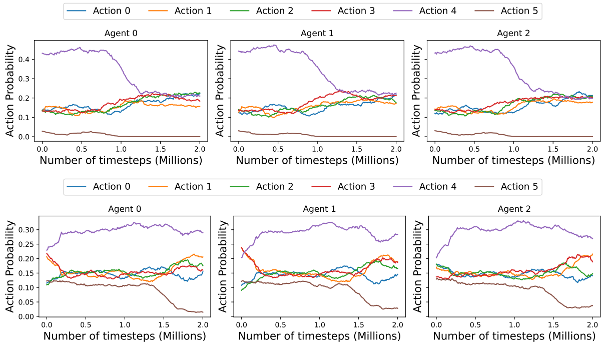


Figure 4: *Task Switching on Foraging-15x15-3p-5f (one seed).* **Top:** MAA2C. **Bottom:** MAPPO.

Sparsity or something else? From figure 2, it would seem that Q-learning methods are almost completely unable to learn in the RWARE setting. (Papoudakis et al. 2021) reasonably hypothesise that this is due to the sparsity of the setting making value decomposition difficult for VDN and QMIX and that IQL is hampered by its limited exploration capabilities. However, both VDN and QMIX have shown a limited capacity to learn some sparse settings (Mguni et al. 2022) and IQL is in some cases able to find policies that can solve highly complex sparse settings in the MARL domain (Mahajan et al. 2019). From figure 5, we can find an alternative explanation for the poor performance of the Q-learning methods. For the high-performing policies found by MAA2C and MAPPO, we can see that there is a high reliance on action 0 which has been revealed by the Task Switching measurement. This indicates that the performance issues are potentially not purely due to sparsity but the Q-learning methods not learning to exploit a specific action. As Q-learning methods use a replay buffer to train from off-policy data, they may not encounter samples exploiting this action in combination with the relevant states frequently enough, making learning the heavily weighted optimal policy difficult. Manually parsing environment trajectories to find this data would be difficult, thus this finding shows how even simple XAI methods can make algorithmic failure points and environment features easier to detect.

Conclusion

This paper highlights the importance of explainability in MARL research. While traditional performance metrics may not fully reveal the true behaviours of agents, we demonstrate how diagnostic tools—specifically, Policy Entropy, Agent Update Divergence, and Task Switching— can provide a more comprehensive overview of agents' behaviour in multi-agent systems. Our work contributes to the growing field of explainable MARL by providing a set of simple yet effective tools that can be applied to various MARL algorithms and environments. Further exploration of explainability in MARL is encouraged to address challenges and fully harness its potential in solving complex problems. It is essential to acknowledge the *limitations* of our work, including the computational overhead that escalates with an increasing number of agents and the potential inefficiency of the Task Switching tool in scenarios with continuous action spaces. Looking ahead, we envision *future research* delving deeper into the application of these tools in more MARL environments. Expanding and refining these diagnostic tools will benefit both researchers and practitioners in the field, pushing the boundaries of explainable multi-agent reinforcement learning and enhancing our ability to tackle complex real-world challenges.

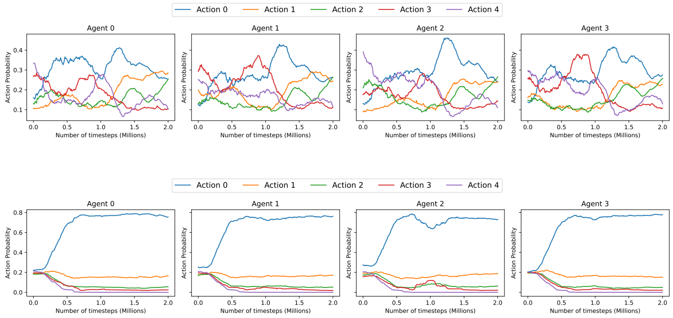


Figure 5: *Comparison of task-switching for MAPPO and IQL algorithms on the RWARE setting* **Top:** IQL. **Bottom** MAPPO.

References

Abdallah, S. 2009. Why Global Performance is a Poor Metric for Verifying Convergence of Multi-agent Learning. *arXiv preprint arXiv:0904.2320*.

Agarwal, R.; Schwarzer, M.; Castro, P. S.; Courville, A.; and Bellemare, M. G. 2022. Deep Reinforcement Learning at the Edge of the Statistical Precipice. arXiv:2108.13264.

Albrecht, S. V.; and Ramamoorthy, S. 2015. A Game-Theoretic Model and Best-Response Learning Method for Ad Hoc Coordination in Multiagent Systems. arXiv:1506.01170.

Albrecht, S. V.; and Stone, P. 2019. Reasoning about Hypothetical Agent Behaviours and their Parameters. arXiv:1906.11064.

Annie, W.; Thomas, B.; Anna, V.; et al. ???? Multiagent Deep Reinforcement Learning: Challenges and Directions Towards Human-Like Approaches. *Artificial Intelligence Review*.

Arrieta, A. B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R.; et al. 2020. Explainable Artificial Intelligence

(XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information fusion*, 58: 82–115.

Canese, L.; Cardarilli, G. C.; Di Nunzio, L.; Fazzolari, R.; Giardino, D.; Re, M.; and Spanò, S. 2021. Multi-agent reinforcement learning: A review of challenges and applications. *Applied Sciences*, 11(11): 4948.

Christianos, F.; Schäfer, L.; and Albrecht, S. 2020. Shared Experience Actor-Critic for Multi-Agent Reinforcement Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 10707–10717. Curran Associates, Inc.

Dazeley, R.; Vamplew, P.; and Cruz, F. 2023. Explainable reinforcement learning for broad-xai: a conceptual framework and survey. *Neural Computing and Applications*, 1–24.

Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Gorsane, R.; Mahjoub, O.; de Kock, R.; Dubb, R.; Singh, S.; and Pretorius, A. 2022. Towards a Standardised Performance Evaluation Protocol for Cooperative MARL. arXiv:2209.10485.

Gronauer, S.; and Diepold, K. 2021. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 55: 895–943.

Heuillet, A.; Couthouis, F.; and Díaz-Rodríguez, N. 2021. Explainability in deep reinforcement learning. *Knowledge-Based Systems*, 214: 106685.

Heuillet, A.; Couthouis, F.; and Díaz-Rodríguez, N. 2022. Collective explainable AI: Explaining cooperative strategies and agent contribution in multiagent reinforcement learning with shapley values. *IEEE Computational Intelligence Magazine*, 17(1): 59–71.

Hu, S.; Xie, C.; Liang, X.; and Chang, X. 2022. Policy Diagnosis via Measuring Role Diversity in Cooperative Multi-agent RL. arXiv:2207.05683.

Kurach, K.; Raichuk, A.; Stańczyk, P.; Zając, M.; Bachem, O.; Espeholt, L.; Riquelme, C.; Vincent, D.; Michalski, M.; Bousquet, O.; and Gelly, S. 2020. Google Research Football: A Novel Reinforcement Learning Environment. arXiv:1907.11180.

Mahajan, A.; Rashid, T.; Samvelyan, M.; and Whiteson, S. 2019. Maven: Multi-agent variational exploration. *Advances in Neural Information Processing Systems*, 32.

Mguni, D. H.; Jafferjee, T.; Wang, J.; Slumbers, O.; Perez-Nieves, N.; Tong, F.; Yang, L.; Zhu, J.; Yang, Y.; and Wang, J. 2022. LIGS: Learnable Intrinsic-Reward Generation Selection for Multi-Agent Learning. arXiv:2112.02618.

Papoudakis, G.; Christianos, F.; Schäfer, L.; and Albrecht, S. V. 2021. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative Tasks. arXiv:2006.07869.

Phan, T.; Ritz, F.; Belzner, L.; Altmann, P.; Gabor, T.; and Linnhoff-Popien, C. 2021. VAST: Value Function Factorization with Variable Agent Sub-Teams.

Puiutta, E.; and Veith, E. M. 2020. Explainable reinforcement learning: A survey. In *Machine Learning and Knowledge Extraction: 4th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2020, Dublin, Ireland, August 25–28, 2020, Proceedings 4*, 77–95. Springer.

Rashid, T.; Samvelyan, M.; de Witt, C. S.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. arXiv:1803.11485.

Roesch, M.; Linder, C. R.; Zimmermann, R. S.; Rudolf, A.; Hohmann, A.; and Reinhart, G. 2020. Smart Grid for Industry Using Multi-Agent Reinforcement Learning. *Applied Sciences*.

Samvelyan, M.; Rashid, T.; de Witt, C. S.; Farquhar, G.; Nardelli, N.; Rudner, T. G. J.; Hung, C.-M.; Torr, P. H. S.; Foerster, J.; and Whiteson, S. 2019. The StarCraft Multi-Agent Challenge.

Scheikl, P. M.; Gyenes, B.; Davitashvili, T.; Younis, R.; Schulze, A.; Müller-Stich, B. P.; Neumann, G.; Wagner, M.; and Mathis-Ullrich, F. 2021. Cooperative Assistance in Robotic Surgery through Multi-Agent Reinforcement Learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1859–1864.

Shapley, L. S. 1953. Stochastic Games*. *Proceedings of the National Academy of Sciences*, 39(10): 1095–1100.

Son, K.; Kim, J.; Ahn, S.; Reyes, R. D. D.; Yi, Y.; and Shin, J. 2022. Disentangling Sources of Risk for Distributional Multi-Agent Reinforcement Learning. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; and Sabato, S., eds., *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 20347–20368. PMLR.

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J. Z.; Tuyls, K.; and Graepel, T. 2017. Value-Decomposition Networks For Cooperative Multi-Agent Learning. arXiv:1706.05296.

Tan, M. 1997. Multi-Agent Reinforcement Learning: Independent versus Cooperative Agents. In *International Conference on Machine Learning*.

Vidhate, D. A.; and Kulkarni, P. 2017. Cooperative multi-agent reinforcement learning models (CMRLM) for intelligent traffic control. In *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, 325–331.

Vouros, G. A. 2022. Explainable deep reinforcement learning: state of the art and challenges. *ACM Computing Surveys*, 55(5): 1–39.

Wang, T.; Dong, H.; Lesser, V.; and Zhang, C. 2020. Roma: Multi-agent reinforcement learning with emergent roles. *arXiv preprint arXiv:2003.08039*.

Whiteson, S.; Tanner, B.; Taylor, M. E.; and Stone, P. 2011. Protecting against evaluation overfitting in empirical reinforcement learning. *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, 120–127.

Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; and Wu, Y. 2022. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35: 24611–24624.

Zhang, H.; and Yu, T. 2020. Taxonomy of reinforcement learning algorithms. *Deep Reinforcement Learning: Fundamentals, Research and Applications*, 125–133.