An Extended Benchmarking of Multi-Agent Reinforcement Learning Algorithms in Complex Fully Cooperative Tasks

George Papadopoulos*
University of Piraeus
Piraeus, Greece
georgepap@unipi.gr

Andreas Kontogiannis*
NTUA & Archimedes AI
Athens, Greece
andreaskontogiannis@mail.ntua.gr

Foteini Papadopoulou
Radboud University
Nijmegen, Netherlands
foteini.papadopoulou@ru.nl

Chaido Poulianou
University of Piraeus
Piraeus, Greece
poulianouxaido@gmail.com

Ioannis Koumentis
University of Piraeus
Piraeus, Greece
iokoumen@unipi.gr

George Vouros
University of Piraeus
Piraeus, Greece
georgev@unipi.gr

ABSTRACT

Multi-Agent Reinforcement Learning (MARL) has recently emerged as a significant area of research. However, MARL evaluation often lacks systematic diversity, hindering a comprehensive understanding of algorithms' capabilities. In particular, cooperative MARL algorithms are predominantly evaluated on benchmarks such as SMAC and GRF, which primarily feature team game scenarios without assessing adequately various aspects of agents' capabilities required in fully cooperative real-world tasks such as multi-robot cooperation and warehouse, resource management, search and rescue, and human-AI cooperation. Moreover, MARL algorithms are mainly evaluated on low dimensional state spaces, and thus their performance on high-dimensional (e.g., image) observations is not well-studied. To fill this gap, this paper highlights the crucial need for expanding systematic evaluation across a wider array of existing benchmarks. To this end, we conduct extensive evaluation and comparisons of well-known MARL algorithms on complex *fully* cooperative benchmarks, including tasks with *images* as agents' observations. Interestingly, our analysis shows that many algorithms, hailed as state-of-the-art on SMAC and GRF, may underperform standard MARL baselines on fully cooperative benchmarks. Finally, towards more systematic and better evaluation of cooperative MARL algorithms, we have open-sourced PyMARLzoo+, an extension of the widely used (E)PyMARL libraries, which addresses an open challenge from [46], facilitating seamless integration and support with all benchmarks of PettingZoo, as well as Overcooked, PressurePlate, Capture Target and Box Pushing.

KEYWORDS

Fully Cooperative Multi-Agent Reinforcement Learning, Benchmarking, Image-based Observations, Open-Source Framework

ACM Reference Format:
George Papadopoulos, Andreas Kontogiannis, Foteini Papadopoulou, Chaido Poulianou, Ioannis Koumentis, and George Vouros. 2025. An Extended Benchmarking of Multi-Agent Reinforcement Learning Algorithms in Complex Fully Cooperative Tasks. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 10 pages.

1 INTRODUCTION

In *fully* cooperative Multi-Agent Reinforcement Learning (MARL) problems, the goal is to train learnable agents in order to maximize their shared cumulative reward, through excessive coordination, sharing of tasks, collaborative exploration, with appropriate decisions for timing and action, and sharing of capabilities. Fully cooperative MARL is of remarkable interest, as it can naturally model many real-world applications, including multi-robot collaboration [5] and warehouse [34], search and rescue [37], human-AI coordination [6], air traffic management [23], logistics networks [27], and supply-chain optimization [21]. Recently, cooperative MARL algorithms are mainly evaluated in settings where adversarial non-learnable agents that interact with the learnable cooperative ones exist: This has received a surge of approaches and methodologies, e.g., see [19, 29, 45, 56], also drawing motivation from multiplayer video-games and team sports.

Despite recent efforts [4, 18, 34, 58] aiming to provide a comprehensive understanding of standard cooperative MARL algorithms' capabilities through benchmarking, MARL evaluation still lacks systematic diversity and reliability for the following reasons:

- Most state-of-the-art (SoTA) cooperative MARL algorithms are predominantly evaluated, and possibly overfit, as pointed out in [14], on specific cooperative-competitive benchmarks where a team of cooperative learning agents competes against a team of bots with fixed policies, namely SMAC [10, 41] and GRF [25]. However, we argue that these benchmarks do not allow adequate evaluation of subtle issues involved in *fully* cooperative MARL, including: excessive coordination and exploration capabilities, sharing of capabilities, appropriate timing in the execution of actions, complimentary observability, scaling to large numbers of agents, and possibly with sparse rewards. For instance, tasks from the LBF benchmark [34],and particularly those with large grids, effectively capture more diverse requirements of fully cooperative multi-robot collaboration: They require agents to first engage in *extensive joint exploration* to identify a *certain* food target, followed by coordinated joint actions where all

[*] Equal Contribution.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

agents must *simultaneously* consume that target. We conjecture that these aspects are crucial for fully cooperative, real-world tasks. On the other hand, these benchmarks emphasize developing skills that do not play dominant roles for fully cooperative tasks, such as countering the opponent team (e.g., surviving enemy attacks in SMAC or tackling opponents and preventing goals in GRF).

- Most MARL algorithms are evaluated solely on tasks with low-dimensional (mostly tabular) state spaces, and thus their effectiveness on real-world, high-dimensional, image-based observations has not been studied.
- MARL evaluation does not often report the training times of the proposed algorithms, so the results cannot be interpreted as a function of the compute budget used [14].

To address the above challenges, the main contributions of this paper are the following: **(1)** Our paper investigates the effectiveness of established MARL algorithms and contributes a comprehensive, **updated empirical evaluation and comparison** of MARL algorithms, including algorithms that have demonstrated SoTA performance in SMAC and GRF, across a wide array of complex **fully cooperative** benchmarks. **(2)** To our knowledge, our work is the first to include tasks with *images* representing *high-dimensional observations* in MARL benchmarking. **(3)** We contribute an open-source Python MARL framework, namely *PyMARLzoo+* [1], which extends the (E)PyMARL frameworks [34, 41] (widely used in developing established MARL algorithms, such as [19, 26, 38, 49, 50]), facilitating seamless integration with all PettingZoo tasks, thus addressing an open challenge from [46] (EPyMARL supports only the MPE PettingZoo tasks, which were already integrated in [34]). In addition to the already integrated MPE [30, 31], LBF [3] and RWARE [34] benchmarks, our PyMARLzoo+ also integrates the complex fully cooperative Overcooked [6], PressurePlate [1, 15], Capture Target [33, 52] and BoxPushing [53, 54] benchmarks. **(4)** Our work provides benchmarking of tasks that are indicative of a wide array of real-world applications, and of the evaluation of diverse requirements for fully cooperative MARL, in terms of joint exploration and coordination. **(5)** Our experimental findings demonstrate that many algorithms, which have been SoTA in SMAC and GRF, may underperform standard MARL algorithms in fully cooperative MARL tasks; thus validating the possible overfitting issues of the current MARL evaluation highlighted in [14]. Furthermore, we point out fully cooperative tasks that are very hard to solve with the existing SoTA methods. **(6)** Our benchmarking is the first to report the training times of algorithms as a further measure of the algorithm's performance, so that the reported results are also interpreted as a function of the compute budget used.

2 PRELIMINARIES

2.1 Dec-POMDPs

Fully cooperative MARL is formulated as a Dec-POMDP [32]. A Dec-POMDP for an N-agent task is a tuple $\langle S, A, P, r, F, O, N, \gamma \rangle$, where S is the state space, A is the joint action space $A = A_1 \times \cdots \times A_N$, $P(s' \mid s, a) : S \times A \rightarrow [0, 1]$ is the state transition function, $r(s, a) : S \times A \rightarrow \mathbb{R}$ is the *shared* reward function and $\gamma \in [0, 1)$ is the discount factor. Assuming partial observability, each agent at time step t does not have access to the full state, yet it samples observations $o^i_t \in O_i$ according to the observation function $F_i(s) : S \rightarrow O_i$. The joint observations are denoted by $o \in O$ and are sampled according to $F = \prod_i F_i$. The action-observation history for agent i at time t is denoted by $h^i_t \in H_i$, which includes action-observation pairs until t-1 and o^i_t, on which the agent can condition its individual stochastic policy $\pi^i_{\theta_i}(a^i_t \mid h^i_t) : H_i \times A_i \rightarrow [0, 1]$, parameterised by θ_i. The joint policy is denoted by π_θ, with parameters $\theta \in \Theta$. The objective is to find an optimal joint policy which satisfies the optimal value function $V^*(s) = \max_\theta \mathbb{E}_{a \sim \pi_\theta, s' \sim P(\cdot|s,a), o \sim F(s)} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]$.

2.2 Assumptions of interest

In addition to *partial observability*, in our benchmark analysis we consider and study the following assumptions of interest: (a) we utilize the celebrated *CTDE* MARL schema [30], which has been widely adopted by the cooperative MARL community [2, 14] as it enables conditioning approximate value functions on privileged information in a computationally tractable manner, (b) we evaluate *fully cooperative tasks*, that is, tasks where there are no adversarial non-learnable agents (e.g., bots), or teams of opponent agents, interacting with the learnable agents, (c) we also evaluate complex tasks with *sparse-reward settings* that require excessive joint exploration and cooperation to be solved, (d) we also study tasks with *image-based, high-dimensional state and observation spaces* (as in real-world tasks), which have been very frequent in single-agent RL but not in MARL evaluation, and (e) as in [34, 58] we assume that agents *lack explicit communication channels during execution.*

3 ALGORITHMS

In our benchmark analysis, we evaluate and compare a wide range of well-known MARL algorithms (see Table 1). The selection of these algorithms is based on the following important (but not all-encompassing) criteria: (1) they have been widely used as competitive baselines by the MARL community in recent works, (2) they have achieved SoTA performance in cooperative-competitive MARL benchmarks, such as SMAC and GRF, and in our analysis we aim to test their performance in fully cooperative tasks, (3) they are based on improving joint exploration in sparse-reward tasks, and (4) they present diversity in the RL optimization part (e.g., being value-based or actor-critic based, utilizing standard recurrent architectures or transformer-based).

4 FULLY COOPERATIVE MARL BENCHMARKS

In this section, we highlight complex, partial observable, fully cooperative tasks from eight MARL benchmarks that we believe to be of significant interest for MARL research. These tasks represent a wide range of real-world applications and test diverse requirements for fully cooperative MARL (see also Table 2).

4.1 PettingZoo

PettingZoo [46] is a Python library consisting of several MARL tasks. In our analysis, we utilize the following fully cooperative PettingZoo benchmarks: *Entombed Cooperative*, along with *Pistonball* and *Cooperative Pong*. These tasks allow challenging scenarios that provide useful testbeds for fully cooperative MARL, using

[1] The source code is available at: https://github.com/AILabDsUnipi/pymarlzooplus

Algorithm	Evaluated in	On/Off-Policy	RL Optimization	Network Architectures	Intrinsic Exploration
QMIX [38]	SMAC [41], GRF [25], MPE [31], LBF [3], RWARE [34]	Off-policy	Value-based	RNN & MLP	No
MAA2C [34]	SMAC [41], GRF [25], MPE [31], LBF [3], RWARE [34]	On-policy	Actor-Critic	RNN & MLP	No
COMA [11]	SMAC [41], MPE [31], LBF [3], RWARE [34]	On-policy	Policy Gradient	RNN & MLP	No
MAPPO [57]	SMAC [41], GRF [25], MPE [31], LBF [3], RWARE [34]	On-policy	Actor-Critic	RNN & MLP	No
QPLEX [49]	SMAC [41]	Off-policy	Value-based	RNN & MLP	No
HAPPO [24]	SMAC [41], MA MuJoCo [35]	On-policy	Actor-Critic	RNN & MLP	No
MAT-DEC [51]	SMAC [41], GRF [25], MA MuJoCo [35]	On-policy	Actor-Critic	Transformer & MLP	No
EMC [59]	SMAC [41]	Off-policy	On top of QPLEX	RNN & MLP	*Yes*: curiosity-driven
MASER [20]	SMAC [41]	Off-policy	On top of QMIX	RNN & MLP	*Yes*: subgoal generation
EOI [22]	GRF [25], MAgent [60]	Off-policy	On top of MAA2C	RNN & MLP	*Yes*: individuality
CDS [26]	SMAC [41], GRF [25]	Off-policy	On top of QPLEX	RNN & MLP	*Yes*: diversity & information sharing

Table 1: Summary of the selected MARL algorithms

high-dimensional, RGB-image-based observation spaces. Indicative references to the PettingZoo benchmark include [7, 44, 46–48].

4.1.1 Entombed Cooperative. Here, agents must extensively coordinate to progress as far as possible into a procedurally generated maze. Each agent needs to quickly navigate down a constantly evolving maze, where only part of the environment is visible. If an agent becomes trapped, they lose. Agents can easily find themselves in dead-ends, only escapable through rare power-ups. A major challenge is that optimal coordination requires agents to position themselves on opposite sides of the map, as power-ups appear on one side or the other but can be used to break through walls symmetrically.

4.1.2 Pistonball. Pistonball is a physics-based fully cooperative game in which the goal is to move a ball to the left boundary of the game area by controlling a set of vertically moving pistons. The main challenge lies in achieving highly coordinated, emergent behavior to optimize performance in the environment. Pistonball uses a realistic physics engine, comparable to the game Angry Birds, adding further complexity to the required agent coordination.

4.1.3 Cooperative Pong. Cooperative Pong is a fully cooperative variant of the classic Pong game where two agents control paddles on opposite sides of the screen, aiming to keep the ball in play. The challenge lies in the asymmetry between the agents—particularly the right paddle, which has a more complex tiered shape—and their limited observation space, restricted to their own half of the screen. This setup requires agents to coordinate excessively and develop cooperative strategies to maximize play time.

4.2 Overcooked

Overcooked [6] is a fully cooperative MARL benchmark, which has been developed to address human-AI coordination. The objective is to deliver soups as quickly as possible, with agents required to place up to three ingredients in a pot, wait for the soup to cook, and then deliver it. The tight space introduces significant challenges in terms of coordination, as agents must split tasks on the fly, avoid collisions, and coordinate effectively in order to achieve high reward. In addition, the rewards are sparse, making the environment even more challenging. Indicative references that use this benchmark include [7, 44, 46]. We utilize the following three different layouts.

4.2.1 Cramped Room. Agents operate in a cramped room, that is, a small room without obstacles. This layout focuses on the agents' ability to maneuver in a limited space, requiring precise navigation to avoid physical collisions with the other agent.

4.2.2 Asymmetric Advantages. Agents operate in an asymmetric room, where each agent works in its own distinct space. In this layout, the challenge lies in recognizing and exploiting the differing strengths of each agent, such as speed or access to certain kitchen resources. This task tests the agents' ability to adapt their strategies to complement each other's capabilities, ensuring that each player's strengths are used effectively to enhance overall team performance.

4.2.3 Coordination Ring. Agents operate in a room with a ring. They must coordinate their actions to collect onions from the bottom left corner of the room, make soups in the center left, and deliver the dishes to the top right corner of the ring. This task is the most challenging among the three.

MARL Benchmark	**Interesting Challenges**	**Insights for Real-world Applications**
Entombed Cooperative (PettingZoo)	- RGB observations - dead-ends - excessive coordination to go to opposite sides in order to break through walls symmetrically	- exploration in space missions - search & rescue - collective wildfire movement
Pistonball (PettingZoo)	- RGB observations - excessive coordination for emergent behavior and synchronization	- assembly line coordination - dance and performance choreography
Cooperative Pong (PettingZoo)	- RGB observations - excessive coordination due to asymmetry between the agents	- collaborative video games - performing arts - sports collaboration
Overcooked (*Cramped Room*, *Asymmetric Advantages*, *Coordination Ring*)	- sparse reward - delivering tasks as quickly as possible - agents operating in a confined space - agent collision avoidance - splitting tasks on the fly	- kitchen automation - urgent multi-robot tasks in confined spaces
Pressure Plate	- sparse reward - excessive exploration and coordination to sequentially unlock each room	- multi-robot collaboration - escape-rooms-like tasks - team-based problem solving
Spread (MPE)	- agent collision avoidance - optimal landmark coverage - very complex if N is large	- traffic management - distributed sensor networks - urban planning
LBF	- sparse reward - exploration to identify the food target - excessive coordination to eat the target simultaneously	- multi-robot collaboration - resource management in supply chains - disaster response coordination
RWARE	- sparse reward and high-dimensional observations - excessive coordination to execute a specific sequence of actions, at the right time, without immediate feedback	- robot warehousing - logistics management

Table 2: Summary of the fully cooperative, partially observable, benchmark tasks

4.3 PressurePlate

PressurePlate [1] is a fully cooperative environment, with sparse-reward settings, set within a 2D grid-world composed of multiple locked rooms that can be unlocked when an agent stands on the corresponding pressure plate, culminating in a final room containing a goal chest. The primary challenge for agents lies in effectively coordinating their movements and positions to sequentially unlock each room and ultimately reach the chest. Indicative references that use this benchmark task include [1, 15].

4.4 Multi-agent Particle Environment (MPE)

In our analysis, we utilize the well-known *Spread* task of the MPE benchmark [30] that focus on effective navigation of particle agents. Indicative references that use this benchmark task include [9, 34, 39, 61]. In this task, N agents must cover N landmarks while avoiding collisions. Agents are rewarded for staying close to landmarks and penalized for collisions, creating a trade-off between coordination and collision avoidance. The task becomes more complex as N increases. Compared to [34], the evaluated tasks are more challenging by using more agents and fewer training steps.

4.5 Level Based Foraging (LBF)

Level-Based Foraging (LBF) [3] features fully cooperative grid-world environments where agents, assigned levels, must move and collect food by coordinating their actions. Agents can collect food only if their combined levels meet or exceed the food's level. The main challenge is the sparse rewards, requiring agents to coordinate closely to collect food simultaneously. Unlike previous work [34], we evaluate more complex LBF tasks, even with a larger number of agents, requiring extensive exploration and coordination. Indicative references that use this benchmark include [6, 12, 17, 55].

Benchmark	**Number of Newly Integrated Tasks**
PettingZoo	49
Overcooked	5
Pressure Plate	3
Capture Target	1
Box Pushing	1

Table 3: Newly integrated tasks (in addition to MPE, LBF and RWARE) in PyMARLzoo+.

4.6 Multi-Robot Warehouse (RWARE)

The Multi-Robot Warehouse (RWARE) environment simulates a fully cooperative, partially observable grid-world where agents must find and deliver shelves to workstations. With limited sight and partial observations, agents face challenges such as sparse rewards, only given after successful shelf deliveries. This requires precise action sequences, effective exploration, and excessive agent coordination. Unlike previous work [34], this study evaluates more challenging RWARE tasks in hard mode, demanding efficient exploration and excessive cooperation among agents. Indicative references that use RWARE include [46–48].

4.7 Capture Target

Capture Target [33, 52], a fully cooperative multi-agent single-target task, presents significant challenges, aiming multiple agents to locate and capture a flickering target on a grid. Excessive coordination is essential, as the target is only captured when all agents converge on its location simultaneously, despite limited visibility.

4.8 Box Pushing

Box Pushing [53, 54] is a fully cooperative grid environment where two agents must collaborate to move three boxes—two small and one large—to a target area. The main challenge is that the large box, yielding the highest reward, can only be moved if both agents coordinate by positioning themselves in parallel cells and pushing simultaneously, making precise timing and teamwork essential.

5 PYMARLZOO+: PETTINGZOO, OVERCOOKED, PRESSURE PLATE, CAPTURE TARGET AND BOX PUSHING NOW COMPATIBLE WITH (E)PYMARL

To facilitate a comprehensive understanding of MARL algorithms through benchmarking, we open-source *PyMARLzoo+*, a Python framework which extends the widely used (E)PyMARL [34, 41], providing an integration of many SoTA algorithms on a plethora of existing MARL benchmarks under a common framework. The key features of our framework are presented below:

Newly Integrated Benchmarks. Our PyMARLzoo+ integrates 8 MARL benchmarks described in Section 4. More specifically, as we illustrate in Table 3, in addition to the MPE, LBF and RWARE benchmarks (already integrated in EPyMARL [34]), our framework fully supports all tasks from PettingZoo, along with tasks from Overcooked, Pressure Plate, Capture Target, and Box Pushing.

Newly Integrated Algorithms. Our PyMARLzoo+ integrates all the algorithms described in Section 3. We note that, except for the standard baselines (that is, MAA2C, MAPPO and QMIX) already integrated in EPyMARL, all remaining algorithms were integrated as part of our work. Furthermore, the widely used *Prioritized Replay Buffer* [42] has been integrated into the off-policy algorithms.

Image (Observation) encoding. We incorporate three pre-trained architectures as image encoders to transform image-based observations into tabular format for policy learning. Specifically, we integrate the following options: *ResNet18* [16], *CLIP* [36], and *SlimSAM* [8].ResNet18, commonly used in single-agent RL (e.g., in [43]), can capture spatial hierarchies, which are beneficial for extracting relevant features from complex visual data. CLIP, also used in single agent RL (e.g., in [13]), utilizes natural language supervision to produce robust and semantically meaningful representations, facilitating model generalization across diverse tasks by linking visual inputs with textual descriptions. Similarly, SlimSAM incorporates a streamlined self-attention mechanism for efficient image encoding. In addition, we offer the option of using *standard CNNs* on raw images for policy training from scratch.

6 BENCHMARK ANALYSIS

6.1 Experimental Setup

In our benchmark analysis, we evaluate MARL algorithms on *Entombed Cooperative*, *Pistonball*, *Cooperative Pong*, *Cramped Room*, *Asymmetric Advantages*, *Coordination Ring*, *Pressure Plate*, *Spread*, *LBF* and *RWARE* benchmarks. To ensure a fair comparison of the selected algorithms, we utilize the same number of training timesteps. Specifically, we use 10 million timesteps for MPE and LBF, 40 million for RWARE, 5 million for PettingZoo; 20 million for PressurePlate, 40 million for the Overcooked's *Cramped Room*, and 100 million timesteps for the Overcooked's *Asymmetric Advantages* and *Coordination Ring*. We note that for all PettingZoo tasks, where the observations are RGB images, we have used a pre-trained ResNet18 model as the observation image encoder. Moreover, except HAPPO, we utilize agents to share their policy parameters.

We adopt the experimental setup of [34]: Throughout the training of all algorithms, we conducted 100 test episodes at every 50,000-step interval to evaluate the current policy. During these test episodes, we record the episode returns, i.e., the accumulated rewards per episode, and compute the average return $\frac{1}{N}\sum_{i=1}^{N} R_{t,i,j}$, where $N = 100$ is the number of test episodes, and $R_{t,i,j}$ is the return of the i-th test episode at timestep t of the j-th experiment.

Our primary metric score is the mean of the unnormalized returns received from the best policy in convergence over five different seeds. As best policy in convergence, we use the policy that received the best return of the last 50 test episodes. We present our results using the mean and the 75% confidence interval.

We used the hyperparameter settings of the newly integrated algorithms, QPLEX, HAPPO, MAT-DEC, CDS, EOI, EMC, and MASER, adhering to configurations suggested by the authors of their original papers for challenging tasks. For MAA2C, MAPPO, COMA and QMIX, we employed the default parameters of EPyMARL. This ensures that the evaluation is as consistent as possible with what is reported in the original papers, allowing for a valid comparison between the algorithms and with results already reported. Our choice to forego extensive tuning is further supported by: (a) preliminary results showing *no significant differences* regarding algorithms' performance across tasks, and (b) the fact that our results, are based on the mean returns from the *best policy* across different seeds.

All experiments were conducted using CPUs, except for tasks within the PettingZoo environment, where image inputs required the use of GPUs. For these image-based tasks, GPUs were utilized for both the frozen image-encoder and training CNN components when a pre-trained encoder was not available. Specifically, we employed two "g5.16xlarge" AWS instances, each featuring 64 vCPUs, 256 GB RAM, and one A10G GPU with 24 GB of memory to manage the computational demands of image processing.

Tasks\Algorithms	QMIX	QPLEX	MAA2C	MAPPO	HAPPO	MAT-DEC	COMA	EOI	MASER	EMC	CDS
2s-8x8-3p-2f	0.94 ± 0.09	0.63 ± 0.48	**0.98 ± 0.02**	0.64 ± 0.34	0.00 ± 0.00	0.24 ± 0.22	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
2s-9x9-3p-2f	0.00 ± 0.00	**0.60 ± 0.49**	**0.59 ± 0.38**	0.18 ± 0.35	0.00 ± 0.00	0.34 ± 0.24	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.50 ± 0.50	0.00 ± 0.00
2s-12x12-2p-2f	0.89 ± 0.01	**0.98 ± 0.00**	0.81 ± 0.03	0.77 ± 0.04	0.52 ± 0.26	0.55 ± 0.06	0.03 ± 0.02	0.34 ± 0.23	0.01 ± 0.01	0.87 ± 0.03	0.91 ± 0.02
4s-11x11-3p-2f	**0.08 ± 0.19**	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
7s-20x20-5p-3f	0.01 ± 0.00	0.00 ± 0.00	**0.78 ± 0.02**	0.57 ± 0.18	0.00 ± 0.00	0.29 ± 0.09	0.03 ± 0.01	0.03 ± 0.01	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00
8s-25x25-8p-5f	0.02 ± 0.01	0.01 ± 0.00	**0.52 ± 0.24**	**0.41 ± 0.23**	0.00 ± 0.00	0.03 ± 0.03	0.03 ± 0.00	0.07 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
7s-30x30-7p-4f	0.06 ± 0.05	0.00 ± 0.00	**0.71 ± 0.02**	0.57 ± 0.03	0.00 ± 0.00	0.08 ± 0.06	0.02 ± 0.00	0.04 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00

Table 4: Results in LBF tasks.

Tasks\Algorithms	QMIX	QPLEX	MAA2C	MAPPO	HAPPO	MAT-DEC	COMA	EOI	MASER	EMC	CDS
tiny-2ag-hard	0.00 ± 0.00	0.61 ± 0.45	2.25 ± 0.62	2.91 ± 0.82	1.46 ± 2.06	**6.00 ± 8.49**	0.02 ± 0.00	**6.58 ± 3.92**	0.00 ± 0.00	1.66 ± 0.30	4.00 ± 2.30
tiny-4ag-hard	0.00 ± 0.00	11.73 ± 10.81	6.87 ± 7.12	17.1 ± 5.31	24.07 ± 0.71	**27.85 ± 19.71**	0.02 ± 0.00	12.09 ± 4.59	0.00 ± 0.00	0.00 ± 0.00	**27.37 ± 6.88**
small-4ag-hard	0.01 ± 0.01	0.94 ± 0.63	1.38 ± 1.26	4.14 ± 2.12	**9.69 ± 3.19**	0.00 ± 0.00	0.03 ± 0.00	0.17 ± 0.01	0.02 ± 0.00	0.05 ± 0.00	4.11 ± 0.32

Table 5: Results in RWARE tasks.

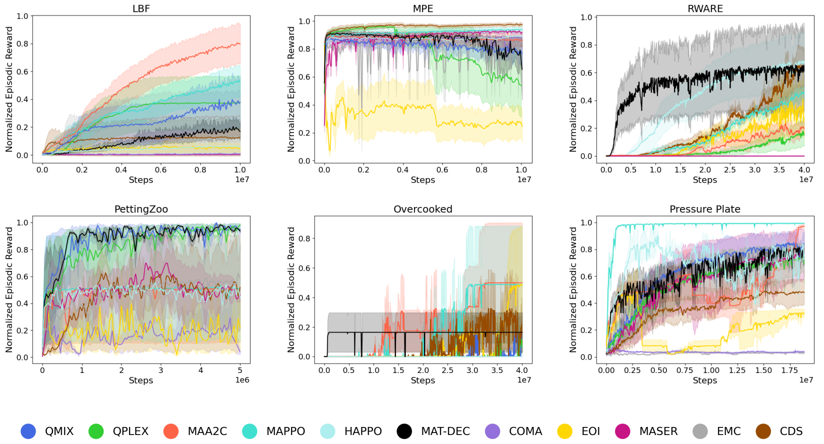

Figure 1: Aggregated Normalized episodic rewards of each benchmark.

Algorithms\Tasks	Spread-4	Spread-5	Spread-8
QMIX	−1278.26 ± 23.13	−2531.17 ± 586.56	−6414.48 ± 27.27
QPLEX	**-766.84 ± 14.39**	−1800.53 ± 194.49	−13260.36 ± 6200.03
MAA2C	−1190.09 ± 99.93	−2312.56 ± 222.08	**-5961.67 ± 66.04**
MAPPO	−971.17 ± 124.22	−1910.20 ± 42.86	**-5926.39 ± 38.48**
HAPPO	−1032.80 ± 45.84	−2000.41 ± 98.20	−6940.61 ± 69.55
MAT-DEC	−1066.62 ± 45.98	−1918.88 ± 15.76	−6843.44 ± 563.39
COMA	−1176.78 ± 33.37	−2003.47 ± 51.18	−6249.07 ± 44.73
EOI	−1963.23 ± 859.27	−5816.66 ± 20.34	−13210.98 ± 5659.53
MASER	−969.06 ± 5.01	−1939.58 ± 113.22	−6242.61 ± 14.40
EMC	−1216.19 ± 10.9	−1961.75 ± 0.71	−6219.14 ± 29.55
CDS	−809.20 ± 47.22	**-1641.77 ± 14.61**	−6250.72 ± 15.44

Table 6: Results in *Spread* (MPE) tasks.

Algorithms\Envs	Cramped Room	Asymmetric Advantages	Coordination Ring
QMIX	0.00 ± 0.00	300.00 ± 300.00	0.00 ± 0.00
QPLEX	86.67 ± 122.57	0.00 ± 0.00	0.00 ± 0.00
MAA2C	**286.80 ± 9.34**	**487.80 ± 107.60**	**0.10 ± 0.10**
MAPPO	**280.00 ± 0.00**	0.30 ± 0.10	**0.07 ± 0.09**
HAPPO	0.00 ± 0.00	160.10 ± 159.9	0.00 ± 0.00
MAT-DEC	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
COMA	0.20 ± 0.16	0.10 ± 0.10	**0.07 ± 0.09**
EOI	**280.0 ± 0.00**	1.60 ± 0.60	**0.13 ± 0.09**
EMC	0.00 ± 0.00	–	–
MASER	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
CDS	186.67 ± 133.00	70.00 ± 70.00	0.00 ± 0.00

Table 8: Results in Overcooked environments.

Algorithms\Envs	Pistonball	Cooperative Pong	Entompted Cooperative
QMIX	**991.59 ± 0.17**	**199.57 ± 0.61**	8.00 ± 0.00
QPLEX	**991.46 ± 0.10**	**197.78 ± 3.14**	8.00 ± 0.00
MAA2C	**990.83 ± 0.04**	−0.33 ± 3.47	6.57 ± 0.05
MAPPO	**990.71 ± 0.10**	13.22 ± 4.61	6.61 ± 0.05
HAPPO	983.60 ± 0.77	25.62 ± 3.36	7.99 ± 0.01
MAT-DEC	982.57 ± 2.44	**200.00 ± 0.00**	10.00 ± 0.00
COMA	678.28 ± 324.06	1.10 ± 7.51	7.68 ± 1.7
EOI	948.35 ± 42.74	−1.64 ± 2.66	6.53 ± 0.02
MASER	**989.39 ± 0.64**	104.25 ± 69.29	8.00 ± 0.00
EMC	265.00 ± 174.58	**196.5 ± 2.83**	8.00 ± 0.00
CDS	415.82 ± 284.37	**197.13 ± 2.26**	**11.10 ± 1.00**

Table 7: Results in PettingZoo using ResNet18.

Algorithms\Envs	4p	6p
QMIX	−210.72 ± 17.84	−3461.77 ± 1020.33
QPLEX	−652.19 ± 9.29	−5183.56 ± 345.46
MAA2C	−281.59 ± 201.77	−547.39 ± 21.00
MAPPO	−135.99 ± 1.32	**-494.08 ± 10.54**
HAPPO	−258.69 ± 224.93	**-584.90 ± 201.68**
MAT-DEC	−876.58 ± 1113.53	−2930.77 ± 3652.53
COMA	−4391.79 ± 108.96	−12360.20 ± 314.06
EOI	−3050.64 ± 1125.83	−9221.16 ± 4796.08
MASER	**-88.44 ± 2.84**	−5257.08 ± 4099.19
EMC	−4518.23 ± 249.68	−12347.60 ± 0.0
CDS	−1926.45 ± 260.85	−8068.23 ± 519.79

Table 9: Results in Pressure Plate tasks.

Algorithms	Episodic Reward	RAM	Training time
MAA2C-ResNet18	**990.83 ± 0.04**	$7GB$	$4d : 15h$
MAA2C-CNN	846.74 ± 19.7	$34GB$	$16d : 0h$

Table 10: Results in PettingZoo's Pistonball comparing ResNet18 as frozen image encoder with trainable CNNs.

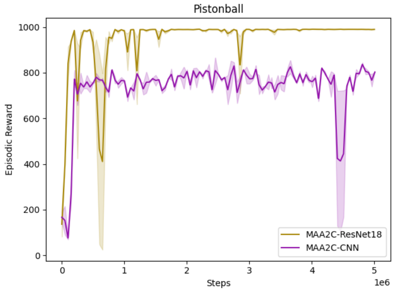

Figure 2: ResNet18 vs Trainable CNNs: MAA2C in PettingZoo's Pistonball task.

6.2 Main Results and Analysis

In this section, we present our main results and the benchmark analysis. We show the main analytical results in Tables 4, 5, 6, 7, 8 and 9, and the averaged main results in Figure 1. To obtain these average results, we use the *normalized* scores for each task of the benchmark and average over these scores. In the tables, we write in bold the values of the best algorithm in a specific task. If the performance of another algorithm was not statistically significantly different from the best algorithm, the respective value is also in bold. Algorithms with green cell color are the best on average in the corresponding benchmark.

6.2.1 Key Findings. Based on results, we highlight the following:

- **Best Algorithms.** The standard MARL algorithms, QPLEX, MAPPO, MAA2C and CDS, demonstrate the most consistent performance across all fully cooperative benchmarks.
- **SoTA exploration-based algorithms mostly underperform.** Exploration-based methods that have reached SoTA performance in SMAC and/or GRF, that is, EOI, EMC and MASER, significantly underperform in most benchmarks compared to the standard methods. In some cases, they even fail entirely, including in tasks with sparse reward settings (e.g., see Tables 4, 5 and 8), in which these methods are expected to improve performance over baselines. The only exception is CDS which is shown to be one of the best algorithms in RWARE and Spread.
- **Value Decomposition.** As demonstrated by the results of QMIX and QPLEX, value decomposition methods tend to converge to suboptimal policies as the number of agents increases, significantly underperforming compared to actor-critic methods (e.g., see the results in Spread, LBF and Pressure Plate). Moreover, our findings contrast with one conclusion of [34] that suggests that value decomposition methods require sufficiently dense rewards to effectively learn to decompose the value function. We show several cases with sparse reward settings, such as in LBF (e.g., *2s-8x8-3p-2f*, *2s-9x9-3p-2f*), RWARE (*tiny-4ag-hard*), and Pressure Plate (4p), where QPLEX, or even QMIX, can successfully converge to effective policies.

6.2.2 Discussion on Algorithms' Performance. Next, we analyse the performance of the evaluated MARL algorithms in the selected fully cooperative benchmark tasks.

***QMIX*, *QPLEX*.** QMIX shows mediocre performance across most tasks, with complete failure in some (e.g., RWARE and many LBF tasks), yet the highest rewards in PettingZoo. Notably, QMIX is the only method to achieve positive rewards in the sparse-reward LBF *4s-11x11-3p-2f* task. In contrast, QPLEX generally outperforms QMIX improving performance in cooperative tasks. However, QPLEX fails in the most challenging LBF and Overcooked tasks and struggles in large-scale scenarios such as *Spread-8* and Pressure Plate's *6p*. Interestingly, despite lacking advanced exploration techniques, QPLEX notably improves over QMIX in RWARE.

***MAA2C, MAPPO, COMA*.** MAA2C is one of the most consistent algorithms across all tasks, showing the best performance in complex sparse-reward LBF tasks and in *Spread-8* (along with MAPPO) and Overcooked. Similarly, MAPPO also ranks as one of the most consistent algorithms but significantly outperforms MAA2C in RWARE, *Spread-5*, *Cooperative Pong*, and Pressure Plate's *6p*. MAPPO achieves the highest rewards on average in Spread (together with CDS) and Pressure Plate. Interestingly, in the RWARE tasks where MAPPO had been the best option in previous work [34], HAPPO, MAT-DEC, and CDS outperform it. Conversely, COMA has one of the lowest performances overall, with competitive results only in *Spread*, *Entombed Cooperative*, and *Coordination Ring*. However, for the last two tasks, all algorithms perform poorly, so none are truly competitive.

***HAPPO, MAT-DEC*.** Both HAPPO and MAT-DEC do not manage to achieve consistent performance across the evaluated benchmarks. More specifically, both methods are shown to be among the best (together with CDS) in RWARE, while HAPPO also achieving good performance in Pressure Plate and MAT-DEC in the high-dimensional PettingZoo tasks. However, both HAPPO and MAT-DEC perform poorly in tasks where efficient exploration and synchronized coordination are crucial, such as in LBF and Overcooked benchmarks.

***MASER, EMC*.** Despite MASER and EMC achieving strong results in sparse-reward SMAC tasks, they perform poorly in most fully cooperative tasks. The notable exception is PettingZoo tasks, where MASER's baseline, QMIX, performs better. MASER also achieves the highest rewards in Pressure Plate (*4p*). Both MASER's and EMC's poor overall performance is attributed to its reliance on Q-values to enhance joint exploration: Since agents rarely receive positive rewards, this leads to misleading intrinsic rewards based on irrelevant Q-value-driven sub-goals. However, Q-values prove to be useful for guiding exploration in LBF's *2s-12x12-2p-2f*, as this task is less sparse but also requires good exploration for success.

***EOI, CDS*.** EOI and CDS outperform MASER and EMC in most tasks. EOI achieves the highest rewards in Overcooked's *Cramped*

Environments\Algorithms	QMIX	QPLEX	MAA2C	MAPPO	HAPPO	MAT-DEC	COMA	EOI	MASER	EMC	CDS
LBF	$0d : 8h$	$0d : 13h$	$0d : 1h$	$0d : 2h$	$0d : 5h$	$0d : 4h$	$0d : 1h$	$0d : 6h$	$0d : 18h$	$2d : 4h$	$0d : 18h$
RWARE	$1d : 22h$	$2d : 17h$	$0d : 9h$	$0d : 12h$	$1d : 1h$	$0d : 20h$	$0d : 9h$	$1d : 18h$	$2d : 16h$	$19d : 1h$	$3d : 2h$
Spread (MPE)	$0d : 15h$	$0d : 21h$	$0d : 2h$	$0d : 3h$	$0d : 9h$	$0d : 6h$	$0d : 2h$	$0d : 11h$	$1d : 12h$	$4d : 9h$	$4d : 9h$
Petting Zoo	$1d : 16h$	$3d : 11h$	$3d : 23h$	$3d : 10h$	$14d : 1h$	$3d : 6h$	$3d : 11h$	$0d : 23h$	$2d : 1h$	$3d : 23h$	$1d : 19h$
Overcooked	$3d : 14h$	$5d : 3h$	$0d : 19h$	$1d : 4h$	$1d : 18h$	$2d : 11h$	$0d : 21h$	$3d : 9h$	$4d : 9h$	$13d : 14h$	$2d : 14h$
Pressure Plate	$0d : 22h$	$1d : 14h$	$0d : 4h$	$0d : 7h$	$0d : 20h$	$0d : 12h$	$0d : 4h$	$0d : 18h$	$1d : 6h$	$9d : 7h$	$2d : 1h$

Table 11: Average (wall-clock) training times over all tasks of each benchmark for all 11 algorithms.

Room (alongside MAA2C and MAPPO) and RWARE's *tiny-2ag-hard*. However, EOI lacks the consistency of the MAA2C backbone algorithm. We attribute this inconsistency to the emphasis on individuality, which can hinder full cooperation and the high level of coordination required in most fully cooperative tasks. In contrast, CDS is more consistent, showing top performance in Spread and RWARE (alongside MAT-DEC and HAPPO) tasks. CDS' improved performance is attributed to the fact that although CDS relies on encouraging agents to be more diverse, as EOI does, it also assures sufficient sharing of the most useful experience of the agents.

6.2.3 Pre-trained Image-Encoder vs Trainable CNNs for Image-based Observations. Our experimental results underscore the effectiveness of employing a frozen pre-trained image encoder, such as ResNet18, over trainable CNNs in multi-agent reinforcement learning (MARL) environments. In the specific case of the PettingZoo's Pistonball task, the use of frozen ResNet18 led to more stable and higher overall policy performance compared to its trainable counterparts. This advantage is clearly illustrated by the smoother convergence curves and the consistently superior performance throughout the training steps, as shown in Figure 2. Moreover, as detailed in Table 10, the non-adaptive nature of frozen ResNet18 highlights its efficiency in managing complex visual contexts without computational overhead. This suggests that pre-trained, static encoders are highly beneficial, providing immediate and reliable performance improvements. The adoption of frozen image encoders significantly improves computational efficiency by eliminating the need for ongoing adjustments during training. This leads to shorter training times for MARL algorithms, as detailed in Table 10. Moreover, by converting image data into compact vector representations, these encoders substantially lower memory requirements, facilitating the execution of complex MARL tasks with RGB-based observations.

6.2.4 Comparison of training times. As can be clearly seen from Table 11, off-policy algorithms generally require longer training due to their use of large replay buffers, which process experiences from multiple past episodes. In contrast, on-policy algorithm, benefit from learning directly from current policy experiences, allowing for faster training. However, in image-based, high-dimensional environments, such as PettingZoo tasks, this parallelism can slow down training, as it does not integrate well with pre-trained models, such as ResNet18, which require GPU resources.

6.2.5 Open Challenges. Based on the results we discussed above, it is evident that fully cooperative MARL tasks need careful algorithmic design, in terms of both excessive coordination and joint exploration, as current SoTA and standard MARL methods are not very effective. Below, we report some significant open challenges arised from our benchmark analysis:
(a) The tasks *Entombed Cooperative* (PettingZoo) and *Coordination Ring* (Overcooked) are the most challenging, as all evaluated algorithms indeed fail to find any effective policy. Any improvement on these tasks would be of remarkable interest.
(b) The sparse-reward LBF tasks, with a large grid, three agents and two foods, the sparse-reward Pressure Plate tasks, with more than 4 agents, and the sparse-reward hard RWARE tasks, with larger grids, are quite challenging, as they require excessive coordinated exploration, and any improvement on these is very interesting.
(c) The Spread tasks, with more than 4 agents are quite challenging, as they require excessive coordination, and any improvement on these without the use of agent communication during execution is very interesting.

7 RELATED WORK

The recent rise in MARL popularity has fragmented community standards and tools, with the frequent introduction of new libraries such as [18, 28, 40, 46]. Among the most popular are PyMARL [41] and EPyMARL [34], both of which have played a crucial role in driving the influx of cooperative MARL algorithms. However, these libraries have somewhat overlooked the integration of fully cooperative MARL environments, pushing researchers to focus on specific benchmarks, such as SMAC [10, 41] and GRF [25], raising concerns about the reliability and generalizability of the proposed algorithms [14]. Despite recent efforts [4, 18, 34, 58] aiming to provide a comprehensive understanding of standard cooperative MARL algorithms through benchmarking, the evaluation of fully cooperative MARL still lacks systematic diversity and reliability.

8 CONCLUSION

In this paper, we highlight and address key concerns in the evaluation of cooperative MARL algorithms by providing an extended benchmarking of well-known MARL methods in fully cooperative tasks. Our extensive evaluations reveal significant discrepancies in the performance of SoTA methods, which can eventually underperform compared to standard baselines. Based on our analysis, as well as by open-sourcing PyMARLzoo+, this paper aims to motivate towards more systematic evaluation of MARL algorithms, encouraging broader adoption of diverse, fully cooperative benchmarks.

ACKNOWLEDGMENTS

The research work contributed by George Papadopoulos was partially supported by the Hellenic Foundation for Research and Innovation (HFRI) under the 5th Call for HFRI PhD Fellowships (Fellowship Number: 20769). The authors also gratefully acknowledge GRNET – National Infrastructures for Research and Technology for providing AWS resources, and the University of Piraeus for funding the participation in the AAMAS 2025 conference and associated travel costs.

REFERENCES

[1] Ibrahim H Ahmed, Cillian Brewitt, Ignacio Carlucho, Filippos Christianos, Mhairi Dunion, Elliot Fosong, Samuel Garcin, Shangmin Guo, Balint Gyevnar, Trevor McInroe, et al. 2022. Deep reinforcement learning for multi-agent interaction. *Ai Communications* 35, 4 (2022), 357–368.

[2] Stefano V Albrecht, Filippos Christianos, and Lukas Schäfer. 2024. *Multi-agent reinforcement learning: Foundations and modern approaches.* MIT Press.

[3] Stefano V. Albrecht and Peter Stone. 2017. Reasoning about Hypothetical Agent Behaviours and their Parameters. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems* (São Paulo, Brazil) *(AAMAS '17)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 547–555.

[4] Matteo Bettini, Amanda Prorok, and Vincent Moens. 2024. Benchmarl: Benchmarking multi-agent reinforcement learning. *Journal of Machine Learning Research* 25, 217 (2024), 1–10.

[5] Wolfram Burgard, Mark Moors, Dieter Fox, Reid Simmons, and Sebastian Thrun. 2000. Collaborative multi-robot exploration. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, Vol. 1. IEEE, IEEE, 476–481.

[6] Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. 2019. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems* 32 (2019).

[7] Changyu Chen, Ramesha Karunasena, Thanh Nguyen, Arunesh Sinha, and Pradeep Varakantham. 2024. Generative modelling of stochastic actions with arbitrary constraints in reinforcement learning. *Advances in Neural Information Processing Systems* 36 (2024).

[8] Zigeng Chen, Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2024. SlimSAM: 0.1% Data Makes Segment Anything Slim. arXiv:2312.05284 [cs.CV] https://arxiv.org/abs/2312.05284

[9] Filippos Christianos, Georgios Papoudakis, Muhammad A Rahman, and Stefano V Albrecht. 2021. Scaling multi-agent reinforcement learning with selective parameter sharing. In *International Conference on Machine Learning*. PMLR, 1989–1998.

[10] Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob Foerster, and Shimon Whiteson. 2024. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems* 36 (2024).

[11] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[12] Elliot Fosong, Arrasy Rahman, Ignacio Carlucho, and Stefano V Albrecht. 2024. Learning Complex Teamwork Tasks using a Given Sub-task Decomposition. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*. 598–606.

[13] Jialu Gao, Kaizhe Hu, Guowei Xu, and Huazhe Xu. 2024. Can pre-trained text-to-image models generate visual goals for reinforcement learning? *Advances in Neural Information Processing Systems* 36 (2024).

[14] Rihab Gorsane, Omayma Mahjoub, Ruan John de Kock, Roland Dubb, Siddarth Singh, and Arnu Pretorius. 2022. Towards a standardised performance evaluation protocol for cooperative marl. *Advances in Neural Information Processing Systems* 35 (2022), 5510–5521.

[15] Nikunj Gupta, Somjit Nath, and Samira Ebrahimi Kahou. 2023. CAMMARL: Conformal Action Modeling in Multi Agent Reinforcement Learning. *arXiv preprint arXiv:2306.11128* (2023).

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[17] Joey Hong, Sergey Levine, and Anca Dragan. 2024. Learning to influence human behavior with offline reinforcement learning. *Advances in Neural Information Processing Systems* 36 (2024).

[18] Siyi Hu, Yifan Zhong, Minquan Gao, Weixun Wang, Hao Dong, Zhihui Li, Xiaodan Liang, Yaodong Yang, and Xiaojun Chang. 2022. Marllib: Extending rllib for multi-agent reinforcement learning. (2022).

[19] Jeewon Jeon, Woojun Kim, Whiyoung Jung, and Youngchul Sung. 2022. Maser: Multi-agent reinforcement learning with subgoals generated from experience replay buffer. In *International Conference on Machine Learning*. PMLR, 10041–10052.

[20] Jeewon Jeon, Woojun Kim, Whiyoung Jung, and Youngchul Sung. 2022. Maser: Multi-agent reinforcement learning with subgoals generated from experience replay buffer. In *International Conference on Machine Learning*. PMLR, 10041–10052.

[21] Chengzhi Jiang and Zhaohan Sheng. 2009. Case-based reinforcement learning for dynamic inventory control in a multi-agent supply-chain system. *Expert Systems with Applications* 36, 3 (2009), 6520–6526.

[22] Jiechuan Jiang and Zongqing Lu. 2021. The emergence of individuality. In *International Conference on Machine Learning*. PMLR, 4992–5001.

[23] Andreas Kontogiannis and George A Vouros. 2023. Inherently Interpretable Deep Reinforcement Learning Through Online Mimicking. In *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*. Springer, 160–179.

[24] Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. 2021. Trust region policy optimisation in multi-agent reinforcement learning. *arXiv preprint arXiv:2109.11251* (2021).

[25] Karol Kurach, Anton Raichuk, Piotr Stańczyk, Michał Zając, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, et al. 2020. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 4501–4510.

[26] Chenghao Li, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. 2021. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems* 34 (2021), 3991–4002.

[27] Xihan Li, Jia Zhang, Jiang Bian, Yunhai Tong, and Tie-Yan Liu. 2019. A Cooperative Multi-Agent Reinforcement Learning Framework for Resource Balancing in Complex Logistics Network. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. 980–988.

[28] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. 2018. RLlib: Abstractions for distributed reinforcement learning. In *International conference on machine learning*. PMLR, 3053–3062.

[29] Bo Liu, Qiang Liu, Peter Stone, Animesh Garg, Yuke Zhu, and Anima Anandkumar. 2021. Coach-player multi-agent reinforcement learning for dynamic team composition. In *International Conference on Machine Learning*. PMLR, 6860–6870.

[30] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems* 30 (2017).

[31] Igor Mordatch and Pieter Abbeel. 2018. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[32] Frans A Oliehoek, Christopher Amato, et al. 2016. *A concise introduction to decentralized POMDPs*. Vol. 1. Springer.

[33] Shayegan Omidshafiei, Jason Pazis, Christopher Amato, Jonathan P How, and John Vian. 2017. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *International Conference on Machine Learning*. PMLR, 2681–2690.

[34] Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. 2021. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative Tasks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS)*.

[35] Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. 2021. Facmac: Factored multi-agent centralised policy gradients. *Advances in Neural Information Processing Systems* 34 (2021), 12208–12221.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[37] Aowabin Rahman, Arnab Bhattacharya, Thiagarajan Ramachandran, Sayak Mukherjee, Himanshu Sharma, Ted Fujimoto, and Samrat Chatterjee. 2022. Adversar: Adversarial search and rescue via multi-agent reinforcement learning. In *2022 IEEE International Symposium on Technologies for Homeland Security (HST)*. IEEE, 1–7.

[38] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research* 21, 178 (2020), 1–51.

[39] Jingqing Ruan, Yali Du, Xuantang Xiong, Dengpeng Xing, Xiyun Li, Linghui Meng, Haifeng Zhang, Jun Wang, and Bo Xu. 2022. GCS: Graph-based coordination strategy for multi-agent reinforcement learning. *arXiv preprint arXiv:2201.06257* (2022).

[40] Alexander Rutherford, Benjamin Ellis, Matteo Gallici, Jonathan Cook, Andrei Lupu, Garðar Ingvarsson, Timon Willi, Akbir Khan, Christian Schroeder de Witt, Alexandra Souly, et al. 2024. JaxMARL: Multi-Agent RL Environments and Algorithms in JAX. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*. 2444–2446.

[41] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. 2019. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043* (2019).

[42] Tom Schaul. 2015. Prioritized Experience Replay. *arXiv preprint arXiv:1511.05952* (2015).

[43] Rutav M Shah and Vikash Kumar. 2021. RRL: Resnet as representation for Reinforcement Learning. In *International Conference on Machine Learning*. PMLR, 9465–9476.

[44] Chapman Siu, Jason Traish, and Richard Yi Da Xu. 2021. Dynamic coordination graph for cooperative multi-agent reinforcement learning. In *Asian Conference on Machine Learning*. PMLR, 438–453.

[45] Jing Sun, Shuo Chen, Cong Zhang, Yining Ma, and Jie Zhang. 2024. Decision-Making With Speculative Opponent Models. *IEEE Transactions on Neural Networks and Learning Systems* (2024).

[46] Jordan Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, et al. 2021. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems* 34 (2021), 15032–15043.

[47] Justin K Terry, Benjamin Black, and Luis Santos. 2020. Multiplayer support for the arcade learning environment. *arXiv preprint arXiv:2009.09341* (2020).

[48] Justin K Terry, Nathaniel Grammel, Sanghyun Son, Benjamin Black, and Aakriti Agrawal. 2020. Revisiting parameter sharing in multi-agent deep reinforcement learning. *arXiv preprint arXiv:2005.13625* (2020).

[49] Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. 2020. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062* (2020).

[50] Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. 2020. ROMA: multi-agent reinforcement learning with emergent roles. In *Proceedings of the 37th International Conference on Machine Learning*. 9876–9886.

[51] Muning Wen, Jakub Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. 2022. Multi-agent reinforcement learning is a sequence modeling problem. *Advances in Neural Information Processing Systems* 35 (2022), 16509–16521.

[52] Yuchen Xiao, Joshua Hoffman, and Christopher Amato. 2020. Macro-action-based deep multi-agent reinforcement learning. In *Conference on Robot Learning*. PMLR, 1146–1161.

[53] Yuchen Xiao, Joshua Hoffman, Tian Xia, and Christopher Amato. 2020. Learning multi-robot decentralized macro-action-based policies via a centralized q-net. In *2020 IEEE International conference on robotics and automation (ICRA)*. IEEE, 10695–10701.

[54] Yuchen Xiao, Weihao Tan, and Christopher Amato. 2022. Asynchronous actor-critic for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems* 35 (2022), 4385–4400.

[55] Mingyu Yang, Yaodong Yang, Zhenbo Lu, Wengang Zhou, and Houqiang Li. 2024. Hierarchical multi-agent skill discovery. *Advances in Neural Information Processing Systems* 36 (2024).

[56] Yaodong Yang, Guangyong Chen, Weixun Wang, Xiaotian Hao, Jianye Hao, and Pheng-Ann Heng. 2022. Transformer-based working memory for multiagent reinforcement learning with action parsing. *Advances in Neural Information Processing Systems* 35 (2022), 34874–34886.

[57] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems* 35 (2022), 24611–24624.

[58] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. 2020. Benchmarking multi-agent deep reinforcement learning algorithms. (2020).

[59] Lulu Zheng, Jiarui Chen, Jianhao Wang, Jiamin He, Yujing Hu, Yingfeng Chen, Changjie Fan, Yang Gao, and Chongjie Zhang. 2021. Episodic multi-agent reinforcement learning with curiosity-driven exploration. *Advances in Neural Information Processing Systems* 34 (2021), 3757–3769.

[60] Lianmin Zheng, Jiacheng Yang, Han Cai, Ming Zhou, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Magent: A many-agent reinforcement learning platform for artificial collective intelligence. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[61] Yifan Zhong, Jakub Grudzien Kuba, Xidong Feng, Siyi Hu, Jiaming Ji, and Yaodong Yang. 2024. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research* 25, 1-67 (2024), 1.