MOSMAC: A Multi-agent Reinforcement Learning Benchmark on Sequential Multi-objective Tasks

Minghong Geng
Singapore Management University
Singapore, Singapore
mhgeng.2021@phdcs.smu.edu.sg

Shubham Pateria
Singapore Management University
Singapore
shubhamp@smu.edu.sg

Budhitama Subagdja
Singapore Management University
Singapore
budhitamas@smu.edu.sg

Ah-Hwee Tan
Singapore Management University
Singapore
ahtan@smu.edu.sg

ABSTRACT

Recent advancements in multi-agent reinforcement learning (MARL) have demonstrated success on various cooperative multi-agent tasks. However, current benchmarks often fall short of representing realistic scenarios that demand agents to execute sequential tasks over long temporal horizons while balancing multiple objectives. To address this limitation, we introduce multi-objective SMAC (MOSMAC), a comprehensive MARL benchmark designed to evaluate MARL methods on tasks involving multiple objectives, sequential subtask assignments, and varying temporal horizons. MOSMAC requires agents to tackle a series of interconnected subtasks in StarCraft II while simultaneously optimizing for multiple objectives, including combat, safety, and navigation. Through rigorous evaluation of nine state-of-the-art MARL algorithms, we demonstrate that MOSMAC presents substantial challenges to existing methods, particularly in long-horizon scenarios. Our analysis establishes MOSMAC as an essential benchmark for bridging the gap between single-objective MARL and multi-objective MARL (MOMARL). The codes for MOSMAC are available at: https://github.com/smu-ncc/mosmac.

KEYWORDS

Multi-agent Reinforcement Learning; Multi-objective Multi-agent Reinforcement Learning; Benchmark

ACM Reference Format:
Minghong Geng, Shubham Pateria, Budhitama Subagdja, and Ah-Hwee Tan. 2025. MOSMAC: A Multi-agent Reinforcement Learning Benchmark on Sequential Multi-objective 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

Multi-agent reinforcement learning (MARL) has demonstrated remarkable success across diverse domains, from traffic signal control [7] to game-playing [31] and stock-trading [1]. These applications predominantly focus on tasks with single *objectives*, such as defeating opponents in RTS games [31] that could be addressed in relatively short action trajectories. A significant gap persists between current MARL testbeds and the requirement of real-world applications, where agents coordinate over long horizons while balancing multiple, often competing objectives.

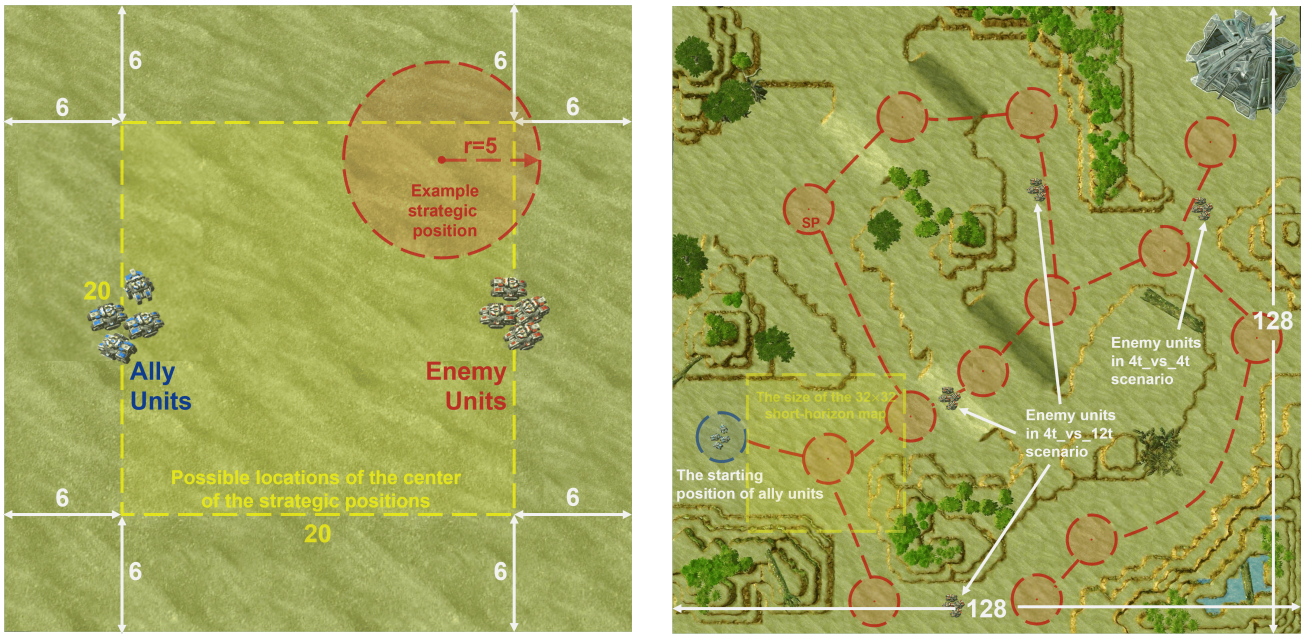
Learning over long horizons presents non-trivial challenges in MARL. As the temporal horizon extends, both multi-agent *exploration* and *temporal credit assignment* become significantly more challenging compared to short-horizon scenarios [15]. Recent theoretical work has also shown that the generalized Rademacher complexity of the hypothesis space for optimal value functions grows with the planning horizon [19], potentially leading to convergence issues and local optima. Despite these known challenges, current MARL literature lacks comprehensive benchmarks for evaluating methods in long-horizon contexts.

This paper introduces multi-objective SMAC (MOSMAC), a comprehensive MARL benchmark that evaluates MARL algorithms on challenging multi-objective MARL (MOMARL) tasks with sequential task allocation. MOSMAC models multi-objective, multi-agent decision-making as a utility-based multi-objective decentralized partially observable Markov decision process (MODec-POMDP). MOSMAC provides two sets of tasks: single-task scenarios, which involve one multi-agent task per episode, and multi-task scenarios, which require completing sequences of multi-agent tasks to reach a final goal. By incorporating complex terrain features — including plains, canyons, ramps, and varying elevations — MOSMAC creates realistic challenges for multi-agent exploration in large state-action space. These features collectively make MOSMAC a uniquely challenging benchmark for evaluating MOMARL algorithms.

Through comprehensive evaluations of nine state-of-the-art MARL algorithms, including IA2C [25], IPPO [32], COMA [9], MAA2C [25], MAPPO [42], IQL [36], MADDPG [21], VDN [35], and QMIX [27], we demonstrate that MOSMAC presents significant challenges, particularly in long-horizon scenarios with multiple sequential tasks. The main contributions of this work are:

(1) We introduce MOSMAC, a new MARL benchmark that challenges MARL with multiple objectives, sequential subtask assignments, and varying temporal horizons.
(2) We bridge the gap between single-objective MARL and multi-objective MARL through a utility-based MODec-POMDP formulation, enabling systematic evaluation of MARL algorithms on multi-objective tasks.

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).



(a) An example of single-task MOSMAC scenarios (4t).

(b) The sequential task allocation of multi-task MOSMAC scenarios.

Figure 1: Examples of MOSMAC scenarios. Figure 1(a) illustrates a single-task scenario named *4t*. The dotted red circle marks the strategic position (SP). The center of SP is randomly located on the central 20×20 area, marked by the dotted yellow square. Figure 1(b) illustrates the multi-task MOSMAC scenarios. Each SP is marked by a circular and connected by navigable pathways.

(3) We conduct a comparative study of nine SOTA MARL algorithms on MOSMAC and show the potential of independent learning on complex multi-task cooperative MOMARL tasks.

The rest of this paper is structured as follows. Section 2 reviews related work, subsequent by background information in Section 3. The proposed MOSMAC benchmark is presented in Section 4. Section 5 introduces the evaluation procedure. Section 6 reports the experiments and results. We report our analysis and findings in Section 7. Section 8 concludes and discusses future extensions.

2 RELATED WORK

2.1 Long Horizon Reinforcement Learning

In episodic reinforcement learning, agents interact with environments through *episodes*, where early actions within an episode can substantially influence subsequent outcomes. *Long-horizon RL* specifically addresses scenarios that require agents to plan over long temporal horizons until episodes end [41] and present unique challenges in exploration and temporal credit assignment. These challenges are particularly acute when rewards are sparse [24], as the signals from early actions diminish exponentially with the horizon length. Recent theoretical analysis has revealed that the generalized *Rademacher complexity* of the hypothesis space of optimal value functions scales with the horizon, leading to the collapse of *action gaps* as the horizon extends [19], making it increasingly difficult to distinguish optimal actions from suboptimal ones. While these challenges have been studied in single-agent contexts [19, 41], their implications for multi-agent systems remain largely unexplored, making long-horizon MARL an important open problem.

2.2 Multi-objective MARL (MOMARL)

Many real-world problems involve multi-objective multi-agent systems (MOMAS) [30], where agents collaborate to perform tasks with multiple, often competing objectives. MOMARL algorithms [13, 16, 23, 38] aim to learn *Pareto fronts*, which could be approximated with *non-dominated sets* of *Pareto optimal* policies. These policies represent optimal trade-offs between objectives, such that improving performance on one objective necessarily results in performance degradation on at least one other objective.

MOMARL approaches can be classified based on how they handle preferences over objectives, which affects their policy learning strategy. Single-policy approaches [22, 29] operate in scenarios where the preferences of objectives are known as a prior. In these cases, agents learn a single policy that optimizes a scalarized value function combining rewards from multiple objectives. Multi-policy approaches [13, 20, 29], however, are designed for situations where objective preferences are not pre-determined. Notably, single-policy approaches can still approximate Pareto fronts through an *outer-loop* approach [13], which trains multiple policies using sampled preferences from a defined preference space.

Despite the increasing relevance of multi-objective optimization in multi-agent systems, MOMARL has received relatively less attention, particularly in its integration with deep reinforcement

learning methods [13]. Given this context, our work bridges single-objective MARL to the MOMARL domain through an outer-loop approach via a single-policy MOMARL framework.

2.3 Existing MARL and MOMARL Benchmarks

Due to the prohibitive cost and complexity of training multi-agent systems in real-world environments, researchers predominantly utilize simulation environments for developing and evaluating MARL algorithms. Various single-objective MARL environments have been introduced in recent years. Many multi-agent grid-world environments evolve as extensions of single-agent predecessors such as MazeBase [34]. Level-based Foraging (LBF) [6] presents a set of gird-world food collection tasks in which agents can cooperate and compete. LBF is commonly implemented with $2-4$ agents on maps with sizes smaller than 15×15 [6]. Robot Warehouse (RWARE) [6] is a partially observable environment with sparse rewards, simulating warehouses in a grid world with robots moving and delivering goods to shelves. RWARE supports scenarios with $1-20$ agents on four sizes ranging from 10×11 to 16×29. Pommerman [28] simulates the Bomberman game on a 11×11 2D grid world, consisting of a set of scenarios with four agents that could either be fully competitive or cooperative in two competing teams. A similar idea of simulating video games in a grid world has been implemented in Overcook-AI [4], which provides five fully cooperative scenarios for human-AI cooperation. Each Overcook-AI game contains two players on a 5×4 or 9×5 map and will last for 400 timesteps.

While grid-world environments offer valuable testbeds for multi-agent learning, it is difficult for them to capture the complexity of continuous states and action space characteristic of many real-world problems. Several environments have emerged to address this limitation, including StarCraft II [40], Multi-particle Environments (MPE) [21], Multi-Agent Mujoco (MAMuJoCo) [26], Google Research Football (GRF) [18], Hanabi [2], and Harvest-gathering [14]. In particular, StarCraft II supports training on both full-game [39] and mini-games [40]. In response to the interest in the micromanagement of agents, i.e., to attain policies that yield optimal unit-level actions, SMAC [31] provides a set of widely adopted benchmark tasks to challenge MARL algorithms. However, a recent study reveals the drawback of SMAC in its lack of stochasticity [8]. Consequently, most SMAC scenarios could be addressed with open-loop policies, which solely rely on time step and agent ID in agents' observations. SMACv2 [8] offers a more challenging benchmark task that introduces stochasticity and improves the diversity of units and scenarios. Another extension of SMAC is the StarCraft Multi-Agent Exploration Challenge (SMAC-Exp) [17], which aims to test MARL algorithms' exploration capability to learn implicit multi-stage tasks and environmental factors as well as micro-control in both offensive and defensive scenarios.

Compared with the single-objective MARL, there exists a significant gap in MOMARL benchmarks that feature complex state and action spaces, partial observability, multiple objectives, multiple agents, and decision-making over long horizons [12]. While some studies have attempted to address this issue by modifying existing single-objective MARL benchmark tasks [13], these solutions remain limited in scope. This work substantially expands upon the preliminary testbed introduced by Geng et al. [10], offering a comprehensive analysis and evaluation of the MOSMAC benchmark with new scenarios and additional objectives. Through this expanded study, we demonstrate MOSMAC's effectiveness as a standardized benchmark for advancing MOMARL research.

3 BACKGROUND

This work studies multi-objective multi-agent decision-making problems formalized as a multi-objective decentralized partially observable Markov decision process (MODec-POMDP). In this section, we present the essential background of the MODec-POMDP framework implemented in MOSMAC. Due to the page limit, we direct interested readers to the survey by Rădulescu et al. [30] for a more detailed MOMARL problem formulation and a recent detailed survey by Hayes et al. [12] for general MORL problems.

3.1 Multi-objective Decentralized Partially Observable Markov Decision Process (MODec-POMDP)

Multi-objective multi-agent decision-making problems could generally be defined as a multi-objective stochastic game (MOSG). A MOSG is formally defined by a tuple $M = (S, \mathcal{A}, T, \mathcal{R})$, with $n \geq 2$ agents and $d \geq 2$ objectives, where S is the state space; $\mathcal{A} = A_1 \times ... \times A_n$ is a set of joint actions of n agents and A_i as the set of actions of agent i; $T : S \times A \times S \rightarrow [0, 1]$ is the probabilistic transition function; $\mathcal{R} = \boldsymbol{R_1} \times ... \times \boldsymbol{R_n}$ are the reward functions and $\boldsymbol{R_i} : S \times A \times S \rightarrow \mathbb{R}^d$ is the vectorized reward function of agent i for each of the d objectives. In some settings, multi-objective multi-agent decision-making problems may encounter limitations such as *partially observable*, where agents cannot access the full environment, and *fully cooperative*, where all agents share the same reward function and learn to optimize the joint *utility* of all agents. Extending from the decentralized partially observable Markov decision process (Dec-POMDP) model from the MARL domain, MOSMAC models a multi-objective decentralized partially observable Markov decision process (MODec-POMDP) [30].

3.2 Utility-based MODec-POMDP

In MODec-POMDP, agent i operates according to a policy $\pi_i : S \times A_i \rightarrow [0, 1]$. For discounted infinite-horizon scenarios, an agent aims to find a policy π_i that maximizes the expected discounted long-term *reward*. The value function of policy π_i is defined as: $V^{\pi_i} = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_{i,t} | \boldsymbol{\pi}, \mu_0]$, where $\boldsymbol{\pi}$ is the joint policy for all agents, μ_0 is the initial state distribution, γ is the discount factor, and $r_{i,t} = \boldsymbol{R_i}(s_t, \boldsymbol{a_t}, s_{t+1})$ denotes agent i's reward for the joint action $\boldsymbol{a_t} \in \mathcal{A}$ taken by all agents at timestep t at state $s_t \in S$ and transit to the next state $s_{t+1} \in S$. Similar to the reward vector $r_{i,t} \in \mathbb{R}^d$, the value function $V^{\pi_i} \in \mathbb{R}^d$ is also vector-valued.

In single-objective MARL problems, agents optimize scalar rewards for single objectives. This approach could extend to MODec-POMDP through *utility functions* (also known as *scalarization functions* [12]). The *utility-based* approach assumes a utility function $u : \mathbb{R}^n \rightarrow \mathbb{R}$ maps the reward vector of multiple objectives into a scalar value. The most widely used utility function is *linear utility function*, which maps the rewards $r_{i,t}$ through equation $u(r_{i,t}) = \sum_{d \in D} w_d r_{i,t,d}$, where D is a set of objectives, w_d is the

Table 1: An overview of the selected scenarios in MOSMAC with siege tank as units. Each scenario could be configured with two objectives: combat and navigation, or three objectives: combat, safety, and navigation.

# of Tasks	Unit Types	Name of Scenarios	# of Ally Units	# of Enemy Units	Environment Features	Observation Space of Movement-related Features	Timesteps Limit
Single Task	Siege Tank	3t	3	3	Plain	9	50
		4t	4	4	Plain	9	50
		8t	8	8	Plain	9	100
		12t	12	12	Plain	9	100
Multiple Tasks (2-7 Tasks per episode)	Siege Tank	4t_vs_4t_large_flat	4	4	Plain	9	500
		4t_vs_4t_large_complex	4	4	Cliff, ramps, and high/low grounds	17	500
		4t_vs_12t_large_flat	4	12	Plain	9	500
		4t_vs_12t_large_complex	4	12	Cliff, ramps, and high/low grounds	17	500

weight for objective $d \in D$ with $\sum_{d \in D} w_d = 1$, and $r_{i,t,d}$ is the reward for objective d in reward vector $r_{i,t}$. MOSMAC models a MODec-POMDP under the *team rewards team utility* (TRTU) paradigm [30], where agents share the *team reward* vectors and collectively optimize a common *team utility*.

Specifically, if the utility function is known before learning and maps every possible outcome of the joint actions into a scalar utility, it transforms the MODec-POMDP into a single-objective Dec-POMDP, which could be subsequently addressed by single-objective MARL methods with a single policy. Therefore, while ground-truth utility functions as *priori* knowledge are not always available [12, 30], it is practicable to define appropriate sets of utility functions for specific reward structures [37], e.g., linear utility functions with different weights, and simultaneously learn multiple distinct policies. These policies could be further utilized to make informed *posteriori* decisions based on situational requirements.

4 THE MOSMAC BENCHMARK

MOSMAC is a MOMARL benchmark that models multi-objective multi-agent problems as a MODec-POMDP [10]. Specifically, it incorporates three key objectives that are critical in real-time strategy (RTS) games: combat, safety, and navigation. MOSMAC consists of two distinct types of scenarios. First, MOSMAC extends existing StarCraft II scenarios [8, 11, 17, 31], introducing multiple objectives while maintaining the original map size and timestep limits from the widely adopted SMAC scenarios. This provides a controlled environment for studying the impact of multiple objectives compared to SMAC scenarios. Second, MOSMAC introduces more challenging scenarios that better reflect real-world complexity, where agents must complete sequences of multi-objective tasks in significantly larger environments. These expanded scenarios feature increased map sizes and extended timestep limits, offering a more realistic simulation of real-world challenges.

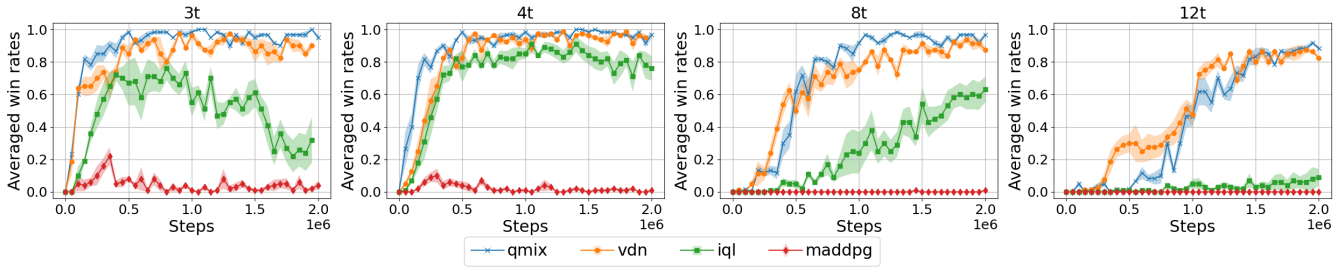
MOSMAC includes three types of units: siege tanks, marines, and stalkers. Their sight and attack ranges remain the same as in the original StarCraft II game. Agents' action space includes moving in one of four directions, attacking a certain enemy, and no-op. The observation space includes the feasibility of moving in one of four directions and the direction and distance toward the strategic position. In scenarios with complex terrains, agents may additionally observe pathing grids and terrain heights to assist learning. Figure 1(a) illustrates the single-task scenarios that are characterized by a group of ally agents, a group of symmetric enemy units controlled by the built-in controller with a difficulty level of 7, and stochastic strategic positions. All agents need to cooperate to arrive at the strategic position when their health value $h \geq 0$ ($h = 0$ marks the agent has been defeated by enemy units) to complete the assigned navigation task. Following SMACv2's design principles, strategic positions are randomized for each episode within a 20×20 region at the center of the 32×32 map, preventing solutions based on open-loop policies [8]. Enemy units are configured to guard the strategic position and engage ally units that approach within the attacking range. As single-task scenarios do not contain complex terrain features, they could be directly compared with existing scenarios in SMAC and SMACv2. The single-task scenarios facilitate direct comparisons with existing SMAC and SMACv2 scenarios.

Figure 1(b) presents the multi-task MOSMAC scenarios implemented on an expanded 128×128 map, available in both flat and complex terrain configurations. MOSMAC defines three distinct task sequences that direct agents from their initial position (marked in blue) to the final destination in the upper right corner of the map. Researchers can also implement advanced path-finding algorithms to generate alternative task sequences for investigating various strategic approaches. In these scenarios, the allied team comprises four units that must contend with either four or twelve enemy units arranged in clusters of four, with a maximum of one enemy cluster encountered during any single episode. Section 4.2 provides a detailed explanation of MOSMAC's sequential task allocation mechanism, while Table 1 summarizes MOSMAC scenarios with siege tank as units.

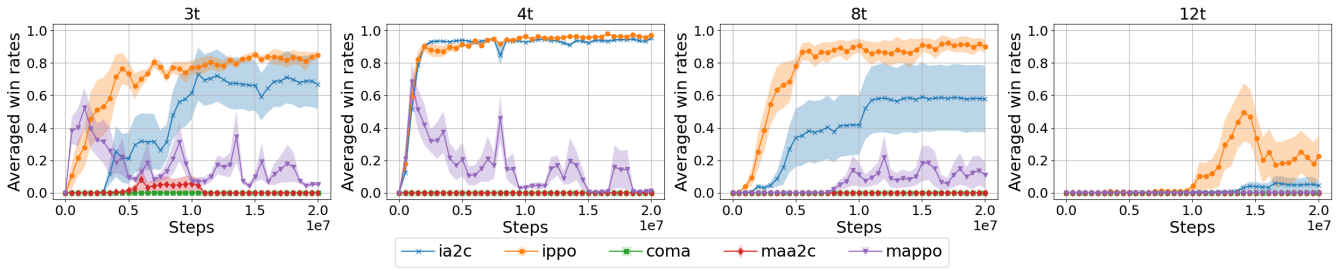
4.1 Multiple Objectives

MOSMAC introduces three critical objectives for the RTS game domain, addressing the key aspects of combat, safety, and navigation. Following the *team reward team utility* paradigm, the reward functions for these objectives are formally defined as:

(1) Objective 1 (combat) [8, 13, 31]: $r_c = \sum_{i=1}^{n} (r_a^i + r_{de}^i)$
(2) Objective 2 (safety) [13]: $r_s = -\sum_{i=1}^{n} (r_h^i + r_{da}^i)$
(3) Objective 3 (navigation): $r_n = \sum_{i=1}^{n} r_r^i$



(a) Off-policy algorithms on single-task MOSMAC scenarios.



(b) On-policy algorithms on single-task MOSMAC scenarios.

Figure 2: The win rates obtained by various MARL algorithms on single-task MOSMAC scenarios.

where r_a^i and r_{de}^i are the rewards for damaging and defeating enemy units by agent i, r_h^i and r_{da}^i are the negative rewards for ally unit i when receiving damage and being destroyed, r_r^i is the reward for reducing the distance towards the strategic position by agent i, and n is the total number of agents.

MOSMAC offers two sets of objectives $D_1 = \{c, n\}$ and $D_2 = \{c, s, n\}$ for various research requirements and complexity levels, where c, s, n stands for combat, safety, and navigation objectives. Given the selected set of objectives D, a utility function could be applied to calculate the team utility. For example, in a tri-objective setting with linear utility function u, the team utility at timestep t is calculated from team reward through $u(r_t) = \sum_{d \in \{c,\, s,\, n\}} w_d r_{t,d}$, where w_d is the weight for objective $d \in D$ with $\sum_{d \in D} w_d = 1$, Besides r_t, agents receive r_w as the terminal reward for winning the game by occupying the strategic position.

MOSMAC provides flexible configuration options for objective combinations, enabling users to customize environments according to their specific research requirements. During benchmark development, we investigated a configuration designated as $D_3 = \{s, n\}$, which considers the safety and navigation objectives. However, our experimental analysis demonstrated that this configuration, lacking the combat objective, produced insufficient reward signals, significantly impeding effective policy learning. Therefore, this paper primarily examines two objective configurations: $D_1 = \{c, n\}$ and $D_2 = \{c, s, n\}$, though the $D_3 = \{s, n\}$ configuration remains available for specialized research purposes.

4.2 Sequential Task Allocation

The sequential task allocation in MOSMAC is formulated as an undirected graph $G = (V, E)$ with n agents traversing a trajectory defined by vertices $\{s, g_1, g_2, ..., g_t\}$, from the start vertex s to the end vertex g_t, where t represents the number of tasks in an episode, as illustrated in Figure 1(a). For each task i, all agents begin at a common start vertex and must collectively navigate to the same target goal vertex g_i. The goal vertex advances to g_{i+1} only when all agents have successfully reached vertex g_i. An episode concludes when all agents arrive at the final vertex g_t. MOSMAC allows researchers to customize the complexity by adjusting the number of tasks t through strategically selecting target positions. In the example shown in Figure 1(b), agents must complete a sequence of 6-8 multi-objective tasks from the initial to the final position. Since this sequential task allocation framework can be simplified to a single-task scenario by setting $t = 1$, it effectively generalizes the single-task scenarios illustrated in Figure 1(a). It is noteworthy that comparable problem formulations have been explored in the *multi-agent pathfinding* (MAPF) [33] and *target assignment and pathfinding* (TAPF) [5] domains, predominantly within grid-world environments. MOSMAC extends these concepts to present a higher-dimensional challenge for MAPF and TAPF methodologies within real-time strategy (RTS) game domains.

5 EVALUATION

5.1 Evaluation Protocol

We compare nine MARL algorithms on the MOSMAC benchmarks with the EPyMARL framework [25]. The algorithms encompass both independent learning approaches, including IA2C, IQL, and IPPO, and centralized *training decentralized execution* (CTDE) methods, including MAA2C, COMA, VDN, QMIX, MAPPO, and MADDPG. The low sample efficacy of on-policy algorithms is compensated by increasing the maximum training steps. In single-task scenarios, on-policy algorithms are trained for 20 million timesteps, while off-policy algorithms are trained for two million time steps.

In scenarios with sequential task allocation, the timesteps for on-policy and off-policy MARL algorithms are 50 million and 10 million, respectively. During training, evaluations of 100 episodes are repeated at constant intervals.

As introduced in Section 3, this study utilizes MARL methods as *single-policy* multi-objective multi-agent decision-making solutions with *linear utility functions*, learning towards one optimal policy with a prescribed utility function. The default preferences for all objectives are $1/n$, where n is the number of objectives considered in the scenario. We further utilized the *outer-loop* approach introduced by Hu et al. [13] to measure MARL methods on MODec-POMDP with a set of different preferences $\mathbf{w}$ to find the Pareto fronts.

5.2 Evaluation Metrics

As a bridge between the MORL and MARL domains, MOSMAC enables the tracking and comparison of performance metrics from both fields. Due to page limits, this paper selectively reports four crucial metrics, two from the MARL domain and two from the MORL domain. For MARL, we present the averaged win rates [31] and average episode returns [25], which are widely used to assess the performance of MARL algorithms. From the MORL perspective, we focus on the *hypervolume* and *sparsity* metrics. In this work, we adopt the definitions of hypervolume and sparsity as described by Basaklar et al. [3]. Hypervolume is a widely used metric in multi-objective optimization that measures the volume of the dominated space in the objective space. A larger hypervolume indicates that the MOMARL algorithm has achieved a more desirable Pareto front approximation. On the other hand, sparsity quantifies the distribution and diversity of the solutions along the Pareto front. A lower sparsity value suggests that the algorithm has found a denser set of solutions, which is preferable as it provides a more comprehensive representation of the possible trade-offs [3]. By considering both hypervolume and sparsity, we can evaluate the quality and diversity of the solutions obtained by MOMARL algorithms in a comprehensive manner.

Hypervolume. The hypervolume measures the (hyper-)volume of the objective space dominated by the policies in an approximate coverage set, relative to a given reference point. Specifically, let P be a Pareto front approximation in an L-dimension objective space and contains N solutions. Let $\mathbf{r_0} \in \mathbb{R}^L$ be the reference point. The hypervolume indicator is defined as:

$$I_H(P) := \Lambda(H(P, r_0)) \tag{1}$$

where $H(P, r_0) = \{\mathbf{z} \in \mathbb{R}^L | \exists 1 \leq i \leq N : \mathbf{r_0} \preceq \mathbf{z} \preceq P_i\}$ with P_i being the i^{th} solution in P and $\preceq$ is the relation operator of multi-objective dominance. $\Lambda(.)$ denotes the Lebesgue measure.

Sparsity. Sparsity quantifies the distance between solutions in the Pareto front, measuring the distribution of the solutions [3, 12, 13]. A common sparsity measurement is the averaged Euclidean distance between consecutive solutions in the Pareto front. The sparsity is mathematically defined as:

$$S_p(P) := \frac{1}{N-1} \sum_{j=1}^{L} \sum_{i=1}^{N-1} (P_{i_j} - P_{(i+1)_j})^2 \tag{2}$$

where P_i is the i^{th} solution in P and $P_{(i+1)_j}$ is the value of solution P_i on the j^{th} objective, given the solutions in P are sorted according to the value on the j^{th} objective.

6 RESULTS

6.1 Results on Single-task MOSMAC

Figure 2 presents the win rates of various MARL algorithms on the single-task MOSMAC scenarios. Among all algorithms, QMIX demonstrates superior performance, achieving the highest results in three out of four scenarios, while MADDPG and MAA2C exhibit substantially lower performance across the test scenarios.

IPPO attains the highest win rates for independent learning approaches in 4t and maintains high win rates in 3t and 8t. However, it encounters challenges when learning effective policies in the more complex 12t scenario. IA2C performs comparably to IPPO in 4t but achieves lower win rates in 3t, 8t, and 12t. IQL successfully reaches 76% win rates in 4t and 63% in 8t but encounters challenges in discovering optimal policies for 3t and 12t.

VDN and QMIX exhibit more consistent performance patterns and successfully converge to optimal policies across all tasks. While VDN demonstrates faster convergence during the initial training phases in the 8t and 12t scenarios, QMIX ultimately converges to more effective policies. MAPPO displays promising results in the early stage of training but experiences significant instability. A notable finding is that independent on-policy learning algorithms significantly outperform their centralized counterparts on single-task MOSMAC scenarios, contrary to observations in many MARL benchmarks. Table 2 summarizes the final win rates, averaged returns, hypervolume, and sparsity metrics across all MARL algorithms evaluated in this study.

A subsequent study conducts a series of experiments with a range of preference weights w to scrutinize agent behaviors. For scenarios with two objectives, we implement five different weights to quantify these effects. As illustrated in Figure 4, it demonstrates that the weighting of objectives presents a significant influence on multi-agent behaviors. Notably, QMIX struggles to learn effective policies under extreme weighting conditions ($w = [0, 1]$ or $w = [1, 0]$), resulting in substantially reduced win rates. In contrast, configurations that assign lower weights to combat objectives while prioritizing navigation objectives demonstrate superior performance. Figure 3 presents the non-dominated set learned by QMIX on the 12t scenario. The full results of all algorithms and scenarios are detailed in the Appendix.

6.2 Results on Multi-task MOSMAC

For multi-task MOSMAC scenarios, we mainly evaluate MARL algorithms on the four scenarios summarized in Table 1. MOSMAC also provides a set of 4t_vs_0t scenarios using the same map while no enemies are provided, presenting a reduced level of complexity. As illustrated in Table 2, multi-task scenarios present substantially greater challenges than single-task ones. While QMIX demonstrates the highest win rates on the 4t_vs_4t_large_flat scenario, all algorithms struggle to achieve consistent convergence. Independent learning algorithms (IQL, IA2C, and IPPO) generally outperform their centralized on-policy counterparts, though with modest win

Table 2: A selective set of results on MOSMAC. The full results are reported in the Appendix.

Metrics	Tasks	Objectives	QMIX	VDN	MADDPG	IQL	IA2C	IPPO	COMA	MAA2C	MAAPO
Win Rates (%)	3t (w=[0.50, 0.50])	c+n	95.00 ± 1.83	90.00 ± 2.74	4.00 ± 2.61	32.00 ± 14.11	66.89 ± 15.15	84.67 ± 2.65	0.00 ± 0.00	0.00 ± 0.00	5.33 ± 0.84
	4t (w=[0.50, 0.50])	c+n	96.67 ± 2.11	95.00 ± 2.24	1.00 ± 0.89	76.00 ± 7.54	95.22 ± 1.14	97.11 ± 0.51	0.00 ± 0.00	0.00 ± 0.00	1.33 ± 0.58
	8t (w=[0.50, 0.50])	c+n	96.67 ± 1.05	87.50 ± 1.93	1.00 ± 0.89	63.00 ± 7.82	57.67 ± 20.45	90.11 ± 4.16	0.00 ± 0.00	0.00 ± 0.00	10.89 ± 8.11
	12t (w=[0.50, 0.50])	c+n	88.33 ± 1.05	82.50 ± 1.11	0.00 ± 0.00	9.00 ± 4.98	4.56 ± 3.58	22.44 ± 12.65	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	12t (w=[0.33, 0.33, 0.33])	c+s+n	11.67 ± 2.36	88.18 ± 7.63	0.00 ± 0.00	0.00 ± 0.00	83.33 ± 2.35	73.30 ± 6.23	0.00 ± 0.00	98.00 ± 2.44	64.00 ± 14.96
	4t_vs_4t_large_flat (w=[0.50, 0.50])	c+n	5.06 ± 3.11	0.00 ± 0.00	0.00 ± 0.00	0.09 ± 0.48	0.00 ± 0.00	0.20 ± 0.45	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	4t_vs_4t_large_complex (w=[0.50,0.50])	c+n	0.71 ± 0.72	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.12 ± 0.36	1.38 ± 1.8	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Averaged Return	3t (w=[0.50, 0.50])	c+n	17.63 ± 2.64	16.92 ± 2.70	7.74 ± 1.88	13.14 ± 5.66	6.22 ± 3.06	16.53 ± 3.63	4.43 ± 0.82	6.03 ± 7.56	9.65 ± 3.58
	4t (w=[0.50, 0.50])	c+n	17.89 ± 1.78	16.92 ± 3.18	10.35 ± 0.97	15.95 ± 4.12	17.5 ± 3.19	17.87 ± 2.76	6.29 ± 0.43	−0.75 ± 2.30	10.15 ± 1.62
	8t (w=[0.50, 0.50])	c+n	17.80 ± 1.42	15.19 ± 4.20	2.17 ± 2.16	13.45 ± 4.43	7.57 ± 1.37	18.38 ± 0.82	6.09 ± 1.53	4.86 ± 1.58	4.55 ± 1.77
	12t (w=[0.50, 0.50])	c+n	17.34 ± 3.43	17.68 ± 3.10	4.22 ± 1.69	8.50 ± 4.79	9.18 ± 2.39	18.75 ± 0.20	4.42 ± 1.90	2.58 ± 1.76	2.27 ± 1.61
	12t (w=[0.33, 0.33, 0.33])	c+s+n	7.76 ± 3.41	11.17 ± 1.21	−5.07 ± 1.16	1.64 ± 4.27	10.21 ± 3.74	9.97 ± 2.53	−4.36 ± 0.98	11.83 ± 0.99	10.31 ± 3.13
	4t_vs_4t_large_flat (w=[0.50, 0.50])	c+n	11.57 ± 3.29	1.51 ± 0.19	0.12 ± 0.7	2.01 ± 0.72	2.04 ± 0.65	7.54 ± 1.52	0.02 ± 0.02	4.93 ± 2.15	2.86 ± 0.29
	4t_vs_4t_large_complex (w=[0.50,0.50])	c+n	7.37 ± 3.44	3.81 ± 1.64	0.3 ± 0.34	4.35 ± 2.24	5.23 ± 1.76	3.42 ± 0.89	−0.33 ± 0.05	6.02 ± 1.45	2.74 ± 0.24
Hypervolume	3t (w=[0.50, 0.50])	c+n	18, 409.26	18, 125.61	17, 332.79	18, 446.17	16, 444.21	18, 776.63	9, 194.87	18, 104.58	18, 792.30
	4t (w=[0.50, 0.50])	c+n	18, 756.31	18, 674.22	17, 291.14	18, 359.50	18, 972.67	19, 436.72	11, 005.12	10, 734.11	18, 484.34
	8t (w=[0.50, 0.50])	c+n	18, 519.78	18, 087.00	7, 246.15	17, 230.55	16, 527.83	19, 135.13	9, 827.89	8, 768.54	10, 150.63
	12t (w=[0.50, 0.50])	c+n	19, 788.61	18, 877.8	6, 591.96	13, 418.29	13, 829.05	19, 044.12	6, 940.52	8, 572.38	9, 390.90
	12t (w ∈ $\{w_1, w_2, w_3, w_4, w_5\}$) (outer-loop)	c+n	19, 801.25	18, 904.51	6, 591.96	14, 998.18	17, 891.24	19, 049.39	6, 940.52	8, 779.10	18, 415.90
	12t (w=[0.33, 0.33, 0.33])	c+s+n	191, 806.79	180, 734.94	0.00	17, 338.72	258, 853.93	278, 112.12	2, 489.88	492, 452.50	235, 512.51
	4t_vs_4t_large_flat (w=[0.50, 0.50])	c+n	21, 959.12	7, 834.61	1, 365.84	8, 779.70	8, 910.24	10, 752.85	506.64	9, 212.61	772.26
	4t_vs_4t_large_complex (w=[0.50,0.50])	c+n	11, 954.10	455.50	0.00	2, 283.81	11, 694.20	12, 030.21	0.00	7, 750.67	658.03
Sparsity	3t (w=[0.50, 0.50])	c+n	143.58	0.54	0.29	1.06	1.37	0.14	0.23	0.91	N.A.
	4t (w=[0.50, 0.50])	c+n	N.A.	127.33	0.35	138.10	N.A.	0.21	0.51	0.31	2.59
	8t (w=[0.50, 0.50])	c+n	1.53	0.37	0.58	0.50	0.47	0.53	0.28	0.59	1.71
	12t (w=[0.50, 0.50])	c+n	0.04	N.A.	0.43	0.23	1.63	0.25	0.20	0.23	0.18
	12t (w ∈ $\{w_1, w_2, w_3, w_4, w_5\}$) (outer-loop)	c+n	0.12	159.5	0.43	0.19	0.32	0.08	0.20	0.31	0.07
	12t (w=[0.33, 0.33, 0.33])	c+s+n	0.67	1.82	9.47	3.65	1.92	0.68	4.15	0.24	0.39
	4t_vs_4t_large_flat (w=[0.50, 0.50])	c+n	1.03	2.03	1.21	2.70	0.26	N.A.	29.00	0.33	0.19
	4t_vs_4t_large_complex (w=[0.50,0.50])	c+n	3.91	2.52	N.A.	N.A.	34.49	0.23	N.A.	0.19	0.47

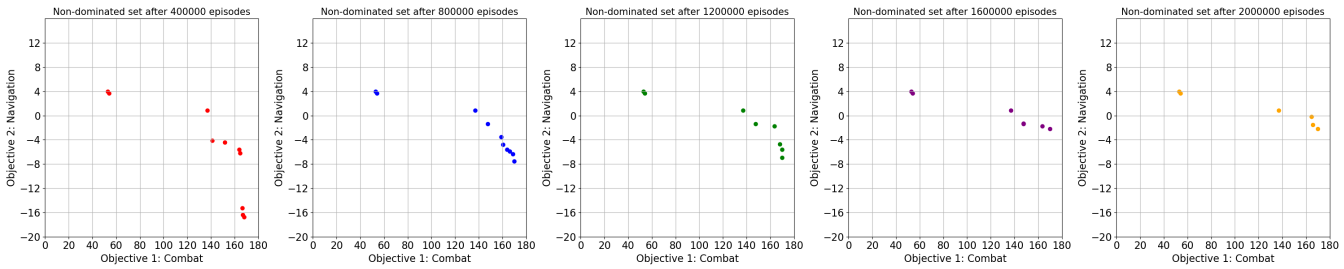


Figure 3: The approximated Pareto front (PF) of QMIX on the 12t scenario. Each PF is generated through the outer-loop approach over five sets of weights. Non-dominated solutions demonstrate a progressive convergence over training.

rates. Notably, independent algorithms also surpass VDN, which had performed comparably to QMIX in single-task scenarios.

On the 4t_vs_4t_large_flat scenario, IPPO achieves the second-highest average return after QMIX, significantly outperforming other methods. From a multi-objective optimization perspective, IPPO, IA2C, and IQL generate comparable Pareto front approximations as measured by hypervolume, slightly outperforming VDN. The introduction of complex terrain features reduces performance disparities between algorithms while increasing overall difficulty. In this more challenging environment, IPPO achieves both higher win rates and hypervolume compared to QMIX. Notably, IQL exhibits lower hypervolume values relative to on-policy methods.

Despite the cooperative nature of the tasks, where all agents must reach the strategic position within a limited temporal horizon, CTDE algorithms do not demonstrate the expected superiority over independent learning approaches. This finding contradicts conventional expectations in cooperative MARL, where centralized training typically provides advantages through shared information. Particularly notable is the superior performance of independent on-policy algorithms in these complex multi-objective scenarios.

The exceptional difficulty of achieving optimal policies in long-horizon MOSMAC tasks, even when facing limited opposition (maximum four enemy units), reveals fundamental challenges in long-horizon MOMARL. We attribute this primarily to imbalanced experience across objectives. In single-task scenarios or enemy-free multi-task scenarios (e.g., 4t_vs_0t), agents encounter balanced objective distributions: either simultaneous combat and navigation signals throughout episodes or exclusively navigation signals. This balance facilitates more effective learning. Conversely, in scenarios like 4t_vs_4t or 4t_vs_12t, agents initially receive predominantly navigation-related signals, with combat-related signals appearing only later and in smaller proportions. This temporal imbalance in objective-related experiences significantly complicates end-to-end learning for MOMARL algorithms. To further investigate this phenomenon, we conducted additional experiments with varying

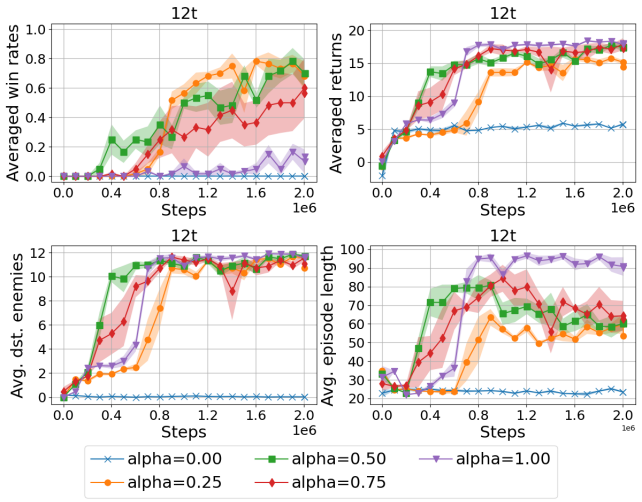


Figure 4: Results of QMIX with 5 sets of weights over two objectives, combat and navigation, on the 12t scenario. Metrics are the average win rates, returns, destroyed enemies and episode lengths. Alpha is the weight for the combat objective.

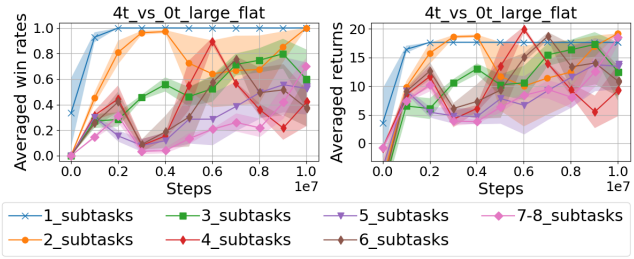


Figure 5: The results of the 4t_vs_0t_large_flat scenario in multi-task MOSMAC with various numbers of subtasks.

horizons and subtasks. As illustrated in Figure 5, the results demonstrate that task difficulty increases progressively with the number of sequential subtasks, highlighting the fundamental challenge of temporal credit assignment in long-horizon MOMARL.

7 ANALYSIS

As a follow-up analysis to the discussion of the effectiveness of independent learning in Papoudakis et al. [25], results in this work reveal that independent learning (IL) algorithms, including IPPO and IA2C, can be more effective than CTDE counterparts, MAA2C and MAPPO, on short-horizon MOSMAC benchmark where each episode provides a randomized multi-objective task. Though centralized learning approaches could avoid the non-stationary issue of IL, it shows that using centralized information during training comes with an expensive cost. That is, as the number of agents increases, learning joint-actions of all agents in on-policy algorithms becomes increasingly difficult. Therefore, training homogeneous agents independently in challenging problems where the roles of agents are interchangeable could be a more efficient approach.

Results also demonstrate the considerable challenge of training MARL algorithms on long-horizon MOSMAC tasks. In long-horizon tasks with explicit subtask completion signals, agents are trained to sequentially complete subtasks, gradually progressing towards the overall task completion. As agents learn each subtask, they adapt their policies to address subsequent challenges, which typically require additional training steps. While one might hypothesize that the training cost for long-horizon tasks would scale linearly with the number of subtasks, our experimental results suggest a non-linear relationship between the training cost of long-horizon tasks and the cumulative cost of their component subtasks. The learning process in multi-task MOSMAC can be conceptualized as training on an expanding set of tasks with progressively increasing complexity. Initially, agents learn to complete scenarios with a single subtask, but the difficulty increases substantially as they encounter scenarios with multiple sequential subtasks once they master the initial subtasks. Furthermore, the sequential nature of these tasks introduces a critical vulnerability: failure at any single subtask can propagate through the sequence, resulting in overall task failure. This cascading failure dynamic adds considerable complexity to the learning process. These observations highlight the importance of developing methods that efficiently learn robust, generalizable policies capable of adapting across the diverse scenarios presented in MOSMAC tasks.

8 CONCLUSION

This paper introduces MOSMAC, a novel MARL benchmark featuring multiple objectives, sequential subtask assignments, and varying temporal horizons. MOSMAC bridges the gap between single-objective and multi-objective MARL through a utility-based MODec-POMDP formulation, enabling systematic evaluation of MARL algorithms on multi-objective tasks. Through a comprehensive evaluation of nine state-of-the-art MARL algorithms, we demonstrate that MOSMAC presents substantial challenges to existing methods. Notably, we found that independent learning approaches outperform centralized training methods when dealing with multi-objective tasks involving homogeneous agents. The insights gained from this work suggest several promising directions for future research. MOSMAC's multi-task framework provides opportunities to evaluate advanced methodologies, particularly in multi-agent exploration and subgoal-based MARL approaches. However, to fully realize the potential of MARL in scaled-up scenarios, future work should focus on developing systematic approaches for multi-task scenario generation and establishing standardized evaluation protocols. Such advancements would facilitate more comprehensive assessment of MARL algorithms across diverse, complex environments.

ACKNOWLEDGMENTS

This research was conducted in collaboration with the DSO National Laboratories, Singapore, and supported in part by the National Research Foundation, Singapore, under its AI Singapore Programme (AISG Award No: AISG2-RP-2020-019) and the Lee Kong Chian Professorship awarded to Ah-Hwee Tan by Singapore Management University.

REFERENCES

[1] Wenhang Bao and Xiao-yang Liu. 2019. Multi-Agent Deep Reinforcement Learning for Liquidation Strategy Analysis. *arXiv preprint arXiv:1906.11046v1 [q-fin.TR]* (June 2019). https://doi.org/10.48550/arXiv.1906.11046

[2] Nolan Bard, Jakob N. Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H. Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, Iain Dunning, Shibl Mourad, Hugo Larochelle, Marc G. Bellemare, and Michael Bowling. 2020. The Hanabi Challenge: A New Frontier for AI Research. *Artificial Intelligence* 280 (March 2020), 103216. https://doi.org/10.1016/j.artint.2019.103216 arXiv:1902.00506 [cs, stat].

[3] Toygun Basaklar, Suat Gumussoy, and Umit Ogras. 2022. PD-MORL: Preference-Driven Multi-Objective Reinforcement Learning Algorithm. https://openreview.net/forum?id=zS9sRyaPFlJ

[4] Micah Carroll, Rohin Shah, Mark K. Ho, Thomas L. Griffiths, Sanjit A. Seshia, Pieter Abbeel, and Anca Dragan. 2019. On the utility of learning about humans for human-AI coordination. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, 5174–5185. https://dl.acm.org/doi/10.5555/3454287.3454752

[5] Yu Quan Chong, Jiaoyang Li, and Katia Sycara. 2024. Optimal Task Assignment and Path Planning using Conflict-Based Search with Precedence and Temporal Constraints. https://doi.org/10.48550/arXiv.2402.08772 arXiv:2402.08772 [cs].

[6] Filippos Christianos, Georgios Papoudakis, Muhammad A. Rahman, and Stefano V. Albrecht. 2021. Scaling Multi-Agent Reinforcement Learning with Selective Parameter Sharing. In *Proceedings of the 38th International Conference on Machine Learning*. Proceedings of Machine Learning Research, Vol. 139. PMLR, 1989–1998. https://proceedings.mlr.press/v139/christianos21a.html

[7] Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. 2020. Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems* 21, 3 (March 2020), 1086–1095. https://doi.org/10.1109/tits.2019.2901791

[8] Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob N. Foerster, and Shimon Whiteson. 2023. SMACv2: An Improved Benchmark for Cooperative Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2212.07489v2 [cs.LG]* (Oct. 2023). https://doi.org/10.48550/arXiv.2212.07489

[9] Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence (AAAI'18/IAAI'18/EAAI'18)*. AAAI Press, New Orleans, Louisiana, USA, 2974–2982. https://dl.acm.org/doi/10.5555/3504035.3504398

[10] Minghong Geng, Shubham Pateria, Budhitama Subagdja, and Ah-Hwee Tan. 2024. Benchmarking MARL on Long Horizon Sequential Multi-Objective Tasks. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS '24)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2279–2281.

[11] Minghong Geng, Shubham Pateria, Budhitama Subagdja, and Ah-Hwee Tan. 2024. HiSOMA: A hierarchical multi-agent model integrating self-organizing neural networks with multi-agent deep reinforcement learning. *Expert Systems with Applications* 252 (Oct. 2024), 124117. https://doi.org/10.1016/j.eswa.2024.124117

[12] Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. 2022. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems* 36, 1 (April 2022), 26. https://doi.org/10.1007/s10458-022-09552-y

[13] Tianmeng Hu, Biao Luo, Chunhua Yang, and Tingwen Huang. 2023. MO-MIX: Multi-Objective Multi-Agent Cooperative Decision-Making With Deep Reinforcement Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 10 (Oct. 2023), 1–15. https://doi.org/10.1109/TPAMI.2023.3283537

[14] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro A. Ortega, D. J. Strouse, Joel Z. Leibo, and Nando de Freitas. 2019. Social Influence as Intrinsic Motivation for Multi-Agent Deep Reinforcement Learning. https://doi.org/10.48550/arXiv.1810.08647 arXiv:1810.08647 [cs, stat].

[15] Nan Jiang and Alekh Agarwal. 2018. Open Problem: The Dependence of Sample Complexity Lower Bounds on Planning Horizon. In *Proceedings of the 31st Conference On Learning Theory*. Proceedings of Machine Learning Research, Vol. 75. PMLR, 3395–3398. https://proceedings.mlr.press/v75/jiang18a.html

[16] Mohamed A. Khamis and Walid Gomaa. 2014. Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework. *Engineering Applications of Artificial Intelligence* 29 (2014), 134–151. https://doi.org/10.1016/j.engappai.2014.01.007

[17] Mingyu Kim, Jihwan Oh, Yongsik Lee, Joonkee Kim, Seonghwan Kim, Song Chong, and Seyoung Yun. 2023. The StarCraft Multi-Agent Exploration Challenges: Learning Multi-Stage Tasks and Environmental Factors Without Precise Reward Functions. *IEEE Access* 11 (2023), 37854–37868. https://doi.org/10.1109/ACCESS.2023.3266652

[18] Karol Kurach, Anton Raichuk, Piotr Stańczyk, Michał Zając, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. 2020. Google Research Football: A Novel Reinforcement Learning Environment. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 04 (April 2020), 4501–4510. https://doi.org/10.1609/aaai.v34i04.5878 Number: 04.

[19] Lucas Lehnert, Romain Laroche, and Harm van Seijen. 2018. On Value Function Representation of Long Horizon Problems. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1 (April 2018). https://doi.org/10.1609/aaai.v32i1.11646

[20] Chunming Liu, Xin Xu, and Dewen Hu. 2015. Multiobjective Reinforcement Learning: A Comprehensive Overview. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 3 (March 2015), 385–398. https://doi.org/10.1109/TSMC.2014.2358639

[21] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Curran Associates Inc., Long Beach, California, USA, 6382–6393. https://dl.acm.org/doi/10.5555/3295222.3295385

[22] Patrick Mannion, Sam Devlin, Jim Duggan, and Enda Howley. 2018. Reward shaping for knowledge-based multi-objective multi-agent reinforcement learning. *The Knowledge Engineering Review* 33 (2018), e23. https://doi.org/10.1017/S0269888918000292

[23] Patrick Mannion, Karl Mason, Sam Devlin, Jim Duggan, and Enda Howley. 2016. Multi-Objective Dynamic Dispatch Optimisation using Multi-Agent Reinforcement Learning: (Extended Abstract). In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, USA, 1345–1346. http://dl.acm.org/citation.cfm?id=2937152

[24] Ian Osband, Benjamin Van Roy, and Zheng Wen. 2016. Generalization and Exploration via Randomized Value Functions. In *Proceedings of The 33rd International Conference on Machine Learning*. PMLR, 2377–2386. https://proceedings.mlr.press/v48/osband16.html ISSN: 1938-7228.

[25] 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*, Vol. 1. Curran Associates Inc. https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/a8baa56554f96369ab93e4f3bb068c22-Abstract-round1.html

[26] Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Boehmer, and Shimon Whiteson. 2021. FACMAC: Factored Multi-Agent Centralised Policy Gradients. In *Advances in Neural Information Processing Systems*, Vol. 34. Curran Associates, Inc., 12208–12221. https://proceedings.neurips.cc/paper/2021/hash/65b9eea6e1cc6bb9f0cd2a47751a186f-Abstract.html

[27] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *arXiv:1803.11485 [cs, stat]* (June 2018). http://arxiv.org/abs/1803.11485

[28] Cinjon Resnick, Wes Eldridge, David Ha, Denny Britz, Jakob Foerster, Julian Togelius, Kyunghyun Cho, and Joan Bruna. 2022. Pommerman: A Multi-Agent Playground. https://doi.org/10.48550/arXiv.1809.07124 arXiv:1809.07124 [cs].

[29] Diederik Marijn Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. 2013. A Survey of Multi-Objective Sequential Decision-Making. *Journal of Artificial Intelligence Research* 48, 1 (Oct. 2013), 67–113. https://doi.org/10.1613/jair.3987

[30] Roxana Rădulescu, Patrick Mannion, Diederik M. Roijers, and Ann Nowé. 2019. Multi-objective multi-agent decision making: a utility-based analysis and survey. *Autonomous Agents and Multi-Agent Systems* 34, 1 (Dec. 2019), 10. https://doi.org/10.1007/s10458-019-09433-x

[31] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. 2019. The StarCraft Multi-Agent Challenge. *arXiv preprint arXiv:1902.04043v5 [cs.LG]* (Dec. 2019). http://arxiv.org/abs/1902.04043

[32] Christian Schroeder de Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip H. S. Torr, Mingfei Sun, and Shimon Whiteson. 2020. Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge? *arXiv preprint arXiv:2011.09533v1 [cs.AI]* (Nov. 2020). https://doi.org/10.48550/arXiv.2011.09533

[33] Roni Stern, Nathan Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Kumar, Roman Barták, and Eli Boyarski. 2019. Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks. *Proceedings of the International Symposium on Combinatorial Search* 10, 1 (2019), 151–158. https://doi.org/10.1609/socs.v10i1.18510 Number: 1.

[34] Sainbayar Sukhbaatar, Arthur Szlam, Gabriel Synnaeve, Soumith Chintala, and Rob Fergus. 2016. MazeBase: A Sandbox for Learning from Games. https://doi.org/10.48550/arXiv.1511.07401 arXiv:1511.07401 [cs].

[35] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. 2017. Value-Decomposition Networks For Cooperative Multi-Agent Learning. *arXiv preprint arXiv:1706.05296v1 [cs.AI]* (June 2017). https://doi.org/10.48550/arXiv.1706.05296

[36] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. 2015. Multiagent Cooperation and Competition with Deep Reinforcement Learning. http://arxiv.org/abs/1511.08779 arXiv:1511.08779 [cs, q-bio].

[37] Peter Vamplew, Cameron Foale, Conor F. Hayes, Patrick Mannion, Enda Howley, Richard Dazeley, Scott Johnson, Johan Källström, Gabriel Ramos, Roxana Radulescu, Willem Röpke, and Diederik M. Roijers. 2024. Utility-Based Reinforcement Learning: Unifying Single-objective and Multi-objective Reinforcement Learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS '24)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2717–2721.

[38] Kristof Van Moffaert, Tim Brys, Arjun Chandra, Lukas Esterle, Peter R. Lewis, and Ann Nowe. 2014. A novel adaptive weight selection algorithm for multi-objective multi-agent reinforcement learning. In *2014 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Beijing, China, 2306–2314. https://doi.org/10.1109/IJCNN.2014.6889637

[39] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (Nov. 2019), 350–354. https://doi.org/10.1038/s41586-019-1724-z

[40] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy Lillicrap, Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, and Rodney Tsing. 2017. StarCraft II: A New Challenge for Reinforcement Learning. *arXiv:1708.04782 [cs]* (Aug. 2017). http://arxiv.org/abs/1708.04782 arXiv: 1708.04782.

[41] Ruosong Wang, Simon S Du, Lin Yang, and Sham Kakade. 2020. Is Long Horizon RL More Difficult Than Short Horizon RL?. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 9075–9085. https://proceedings.neurips.cc/paper/2020/hash/6734fa703f6633ab896eecbdfad8953a-Abstract.html

[42] 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 (Dec. 2022), 24611–24624. https://proceedings.neurips.cc/paper_files/paper/2022/hash/9c1535a02f0ce079433344e14d910597-Abstract-Datasets_and_Benchmarks.html