

Learning Together and Against Each Other: How Multiple Reinforcement-Learning Agents Coordinate, Compete and Where the Field is Heading

Rajiv Ranganath¹; Thrisha R.²; Vismaya M.³; Akshitha Katkeri⁴

^{1,2,3,4} Dept. of Computer Science and Engineering
B.N.M. Institute of Technology
Bengaluru, India

Publication Date: 2026/09/26

Abstract: When several self-directed learners share one environment, each must improve its own behavior while the others are also changing theirs. This is the setting studied by multi-agent reinforcement learning (MARL), and the agents involved may be working toward one goal, pulling in opposite directions, or some combination of the two. In this review we first lay out the mathematical objects used to describe such interaction, namely Markov games and their decentralized, partially observed variants, and then group the algorithmic literature into three lineages: agents that learn in isolation, methods that factor a team value into per-agent pieces (VDN, QMIX and their descendants), and actor-critic schemes that train against a critic with global knowledge (MADDPG, COMA, MAPPO). We discuss the obstacles that distinguish MARL from its single-agent counterpart, namely a moving-target learning problem, the difficulty of dividing a shared reward among team members, limited local views, and growth of the joint action space, and explain why training with global information but acting on local information has become the standard design. Uses of MARL in real-time strategy games, robot teams, automated vehicles, wireless resource sharing, and the orchestration of language-model agents are reviewed alongside the test suites the community relies on. We close by pointing to unresolved questions concerning scale, cooperation with unfamiliar partners, and deployment safety.

Keywords: Multi-Agent Reinforcement Learning; Cooperative Learning; Value Decomposition; Centralized Training Decentralized Execution; Policy Gradient Methods.

How to Cite: Rajiv Ranganath; Thrisha R; Vismaya M; Akshitha Katkeri (2026) Learning Together and Against Each Other: How Multiple Reinforcement-Learning Agents Coordinate, Compete and Where the Field is Heading. *International Journal of Innovative Science and Research Technology*,11(9), 1440-1443. <https://doi.org/10.38124/ijisrt/26sep517>

I. INTRODUCTION

Reinforcement learning (RL) describes how a decision-maker can improve its behavior by interacting with its surroundings and observing the numerical feedback that follows each action; the approach has delivered headline results in single-agent tasks from arcade games to robot manipulation [1]. Many practical problems, however, involve more than one decision-maker. Self-driving cars share roads with other cars, warehouse robots must split work and stay out of each other's way, and market participants must anticipate the moves of rival traders. Situations like these call for multi-agent reinforcement learning (MARL), where a group of learners adapt at the same time inside one common, and usually shifting, environment [2].

Although MARL reuses most of the tools of single-agent RL, it also raises problems with no single-agent analogue. Because every agent keeps updating its policy, the world seen from the viewpoint of any one of them keeps changing, and the fixed-dynamics assumption behind most single-agent convergence results no longer holds. Cooperative tasks add a second complication: if the team receives one collective reward, deciding how much of it each member actually earned is far from straightforward [3].

Even so, the field has moved quickly over the last ten years, helped by deep neural function approximators, the wide adoption of the centralized-training-with-decentralized-execution (CTDE) design, and shared benchmark suites. This progress is reflected in a sequence of influential algorithms, among them Value Decomposition Networks (VDN) [4], QMIX [5], MADDPG [6], COMA [7], and MAPPO [8].

This paper gives a structured account of this literature. Section II introduces the theoretical formalisms used to model multi-agent interaction. Section III presents a taxonomy of MARL settings. Section IV surveys the major algorithm families. Section V discusses core challenges. Section VI surveys application domains. Section VII reviews benchmark environments. Section VIII discusses open problems, and Section IX concludes.

II. THEORETICAL FOUNDATIONS

A. From Markov Decision Processes to Markov Games

The usual model for a single learner is the Markov Decision Process (MDP), in which the aim is a policy that maximizes the expected sum of discounted rewards. With more than one learner, the state that follows and the reward each agent collects are determined by what every agent chose together, not by any single choice. Littman brought the Markov game (also called a stochastic game) into RL to capture exactly this, with transitions and rewards defined over the combined action of all participants [9]. The framework contains both MDPs and repeated matrix games as special cases.

When each agent sees only a private, incomplete view of the world, the model must be extended again, giving the Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [10]. This is the usual formalism for cooperative MARL because it encodes the restriction that an agent can base its decisions only on what it has personally seen.

B. Centralized Training with Decentralized Execution

A design principle running through most of contemporary MARL is centralized training with decentralized execution (CTDE). While learning, an algorithm is free to exploit information that will not exist at deployment, for example the full environment state or what the other agents did. Once deployed, each agent must decide using nothing beyond its own observation history. Nearly all of the methods covered in Section IV (VDN [4], QMIX [5], MADDPG [6], COMA [7], MAPPO [8]) follow this pattern.

III. A TAXONOMY OF MULTI-AGENT SETTINGS

The reward structure of a MARL problem is the usual basis for classifying it, and three classes are commonly distinguished [2], [3]. When every agent receives the same reward, the task is purely cooperative and the group simply tries to drive that shared return as high as possible. When the rewards always add up to a constant, the task is purely competitive (zero-sum), and any improvement for one agent comes at the expense of another. Everything in between is called a mixed, or general-sum, game, where interests partly overlap and partly conflict; most real problems fall here.

This taxonomy has direct algorithmic consequences. Cooperative settings admit value-decomposition and shared-reward policy-gradient methods that would be meaningless in competitive settings, where each agent must instead reason about an adversarial opponent. Mixed settings are the most

challenging to solve in general, typically requiring solution concepts drawn from game theory, such as Nash equilibria, rather than a single optimal joint policy.

IV. ALGORITHMS

A. Independent Learning

The most basic strategy is to let each agent learn on its own with an ordinary single-agent method, folding every other agent into the environment. Independent Q-Learning (IQL) is the direct instance of this idea: every agent maintains a private action-value function over its own observations and actions, an arrangement that traces back to Tan's early study contrasting independent and cooperative learners [11]. The approach costs nothing extra as the team grows, but it ignores the moving-target problem entirely and tends to perform badly on hard coordination tasks with sparse rewards or very large joint action spaces.

B. Value-Decomposition Methods

Value-decomposition approaches tackle the reward-sharing difficulty in cooperative tasks by learning a team action-value function that is assembled from per-agent utilities. In VDN, due to Sunehag et al. [4], the team value is simply the sum of the individual values; this allows training to use the team signal while each agent, at run time, just picks the action that maximizes its own utility. The additive form is simple to optimize but can represent only a narrow class of value functions.

QMIX, from Rashid et al. [5], loosens this constraint by passing the individual utilities through a learned mixing network whose parameters are produced by hypernetworks that read the global state, with the mixing constrained to be monotone in each input. Monotonicity guarantees the individual-global-max (IGM) property, so agents can still act greedily and independently. QMIX is now among the most common cooperative-MARL baselines and performs strongly on the StarCraft Multi-Agent Challenge (SMAC) [12]. QTRAN, by Son et al. [13], learns an unconstrained team value in parallel with a VDN-style factored one and can in principle express strictly more than QMIX, while Qatten [14] inserts multi-head attention into the mixing step.

C. Centralized-Critic Policy-Gradient Methods

Rather than decomposing a value function, one can keep a policy-gradient learner and simply give it a critic that sees everything during training. MADDPG, by Lowe et al. [6], takes this route, adapting Deep Deterministic Policy Gradient so that each agent trains a critic over the observations and actions of every agent, while its actor sees only its own observation. Because the critic is per-agent rather than shared, the method also works in mixed cooperative-competitive tasks with continuous actions.

COMA, from Foerster et al. [7], handles reward sharing inside an actor-critic learner by having a centralized critic compute, for each agent, a counterfactual baseline that marginalizes out that agent's own action, thereby isolating its individual contribution. MAPPO, examined in depth by Yu et al. [8], simply runs the Proximal Policy Optimization method

of Schulman et al. [15] under CTDE. In spite of how little it changes, MAPPO turns out to be an unusually strong baseline, equalling or beating more elaborate methods on SMAC, the Hanabi challenge [16], and the multi-agent particle tasks.

D. Communication and Coordination

Another line of work lets agents exchange messages. Differentiable Inter-Agent Learning (DIAL), by Foerster et al. [17], learns the messaging protocol itself by backpropagating through the communication channel during centralized training. CommNet, by Sukhbaatar et al. [18], takes a different route: a single shared network in which, at every communication round, each agent's hidden state is refreshed with the mean of all agents' hidden states.

V. CORE CHALLENGES

A. Non-Stationarity

Since every agent is revising its policy at the same time, the transition and reward behavior any one agent experiences drifts over time, and the convergence proofs of standard single-agent methods, which assume fixed dynamics, no longer apply [19]. CTDE reduces this effect but does not remove it.

B. Credit Assignment and Scalability

When a cooperative team is paid with a single collective reward, working out how much each member's action contributed is genuinely hard, and a poor answer can produce so-called lazy agents that free-ride on their teammates [3], [4]. Scale is a closely related concern: the number of possible joint actions multiplies with every additional agent, which is why parameter sharing is so common and why frameworks such as EPyMARL, which also supports training without shared parameters, have proved useful [20].

C. Generalization to Novel Teammates

Policies produced by self-play or by training with a fixed set of partners can become tuned to those specific partners. Two research directions respond to this, ad hoc teamwork [21] and zero-shot coordination [22], both asking how an agent can work well with partners it has never met before.

VI. APPLICATIONS

A. Strategy Games

AlphaStar from DeepMind paired deep RL with a population-based training league to reach the top human tier in StarCraft II [23], and OpenAI Five relied on massive self-play to compete with professional Dota 2 teams [24].

B. Robotics, Vehicles, and Networks

MARL has been applied extensively to multi-robot coordination, connected and automated vehicles, and wireless resource allocation in mobile edge computing, where distributed decision-makers must adaptively share limited resources under dynamic, decentralized conditions [25].

C. Coordinating Large Language Model Agents

A recent application of MARL techniques is to the coordination and fine-tuning of teams of large language model (LLM) based agents. CORY fine-tunes two copies of an LLM through a sequential cooperative game in which a pioneer produces an initial answer, an observer refines it, and both receive a shared reward for the overall quality of the final output [26].

VII. BENCHMARKS AND EVALUATION

SMAC provides cooperative unit-control scenarios in which a team of independently acting agents must beat a scripted enemy, and it is now the default proving ground for cooperative methods [12]. The Multi-Agent Particle Environment, first used to test MADDPG [6], is still widely used for quick comparisons. EPyMARL was built so that different algorithm families could be compared fairly under one set of implementation choices [20]. Hanabi tests cooperation under hidden information, forcing agents to signal to one another through their actions alone [16].

VIII. OPEN PROBLEMS AND FUTURE DIRECTIONS

Scalability remains a central concern, since most CTDE algorithms have been demonstrated with tens rather than hundreds or thousands of agents. Generalization to novel teammates and opponents remains largely unresolved outside narrow benchmark settings [21], [22]. Safety and robustness represent a further pressing direction, since MARL policies trained under idealized simulation conditions can be vulnerable to distributional shift at deployment. The application of MARL concepts to coordinate populations of LLM agents [26] raises new questions about how classical notions of reward and credit assignment translate to agents whose action space is natural language.

IX. CONCLUSION

Multi-agent reinforcement learning has matured into a substantial and rapidly developing subfield, driven by the formalization of Markov games, the adoption of CTDE, and algorithm families spanning value decomposition, centralized-critic policy gradients, and explicit communication. These algorithms have demonstrated strong empirical performance across games, robotics, and resource allocation, while standardized benchmarks have made rigorous comparison possible. Fundamental challenges around non-stationarity, credit assignment, scalability, and generalization remain only partially solved, and the extension of MARL to LLM agent coordination suggests continued growth into new application domains.

ACKNOWLEDGMENT

The authors are grateful to B.N.M. Institute of Technology, Bengaluru, for the facilities and encouragement that supported this work.

REFERENCES

- [1]. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
- [2]. L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern. C*, vol. 38, no. 2, pp. 156–172, 2008.
- [3]. A. Oroojlooy and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," *Appl. Intell.*, vol. 53, no. 11, pp. 13677–13722, 2023.
- [4]. P. Sunehag et al., "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proc. Int. Conf. Autonomous Agents Multiagent Syst.*, 2018.
- [5]. T. Rashid, M. Samvelyan, C. Schroeder de Witt, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 4295–4304.
- [6]. R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
- [7]. J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018.
- [8]. C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of PPO in cooperative multi-agent games," in *Adv. Neural Inf. Process. Syst.*, vol. 35, 2022.
- [9]. M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 1994, pp. 157–163.
- [10]. D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of Markov decision processes," *Math. Oper. Res.*, vol. 27, no. 4, pp. 819–840, 2002.
- [11]. M. Tan, "Multi-agent reinforcement learning: Independent versus cooperative agents," in *Proc. Int. Conf. Mach. Learn.*, 1993, pp. 330–337.
- [12]. M. Samvelyan et al., "The StarCraft multi-agent challenge," *arXiv preprint arXiv:1902.04043*, 2019.
- [13]. K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, "QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5887–5896.
- [14]. Y. Yang, J. Hao, B. Liao, K. Shao, G. Chen, W. Liu, and H. Tang, "Qatten: A general framework for cooperative multiagent reinforcement learning," *arXiv preprint arXiv:2002.03939*, 2020.
- [15]. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [16]. N. Bard et al., "The Hanabi challenge: A new frontier for AI research," *Artif. Intell.*, vol. 280, Art. no. 103216, 2020.
- [17]. J. Foerster, I. A. Assael, N. de Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Adv. Neural Inf. Process. Syst.*, vol. 29, 2016.
- [18]. S. Sukhbaatar, A. Szlam, and R. Fergus, "Learning multiagent communication with backpropagation," in *Adv. Neural Inf. Process. Syst.*, vol. 29, 2016.
- [19]. P. Hernandez-Leal, B. Kartal, and M. E. Taylor, "A survey and critique of multiagent deep reinforcement learning," *Auton. Agents Multi-Agent Syst.*, vol. 33, no. 6, pp. 750–797, 2019.
- [20]. G. Papoudakis, F. Christianos, L. Schafer, and S. V. Albrecht, "Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks," in *Proc. Adv. Neural Inf. Process. Syst. Datasets and Benchmarks Track*, 2021.
- [21]. R. Mirsky, I. Carlucho, A. Rahman, E. Fosong, W. Macke, M. Sridharan, P. Stone, and S. V. Albrecht, "A survey of ad hoc teamwork research," in *Proc. Eur. Conf. Multi-Agent Syst.*, 2022.
- [22]. J. Treutlein, M. Dennis, C. Oesterheld, and J. Foerster, "A new formalism, method and open issues for zero-shot coordination," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10413–10423.
- [23]. O. Vinyals et al., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.
- [24]. C. Berner et al., "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.
- [25]. Z. Ning and L. Xie, "A survey on multi-agent reinforcement learning and its application," *J. Autom. Intell.*, vol. 3, no. 2, pp. 73–91, 2024.
- [26]. H. Ma et al., "Coevolving with the other you: Fine-tuning LLM with sequential cooperative multi-agent reinforcement learning," in *Adv. Neural Inf. Process. Syst.*, vol. 37, 2024.