

Review

Advances in Multi-Agent Deep Reinforcement Learning: Methods with Applications and Challenges

Abdur Rakib ^{1,*} , Khoa Phung ² , Marco Perez Hernandez ²  and Mehmet Emin Aydin ² 

¹ Centre for Future Transport and Cities (CFTC), Coventry University, Coventry CV1 5FB, UK

² School of Computing and Creative Technologies, University of the West of England, Bristol BS16 1QY, UK; khoa.phung@uwe.ac.uk (K.P.); marco.perezhernandez@uwe.ac.uk (M.P.H.); mehmet.aydin@uwe.ac.uk (M.E.A.)

* Correspondence: ad9812@coventry.ac.uk

Abstract

Multi-agent deep reinforcement learning (MARL) extends deep reinforcement learning (DRL) to environments involving multiple interacting agents and has enabled applications in domains such as autonomous vehicles, robotics, unmanned aerial vehicles (UAVs), and multi-player games. Compared with single-agent learning, MARL introduces additional challenges, including non-stationarity, partial observability, multi-agent credit assignment, and scalability. This paper presents a narrative survey of recent developments in MARL and discusses major approaches proposed to address these challenges. In particular, we examine research directions centred on centralised training with decentralised execution (CTDE), value decomposition, learned communication, graph-based methods, and model-based learning. We further discuss commonly used benchmark environments and evaluation practices, highlighting considerations related to reproducibility, robustness, and generalisation. Finally, we outline open research challenges and future directions concerning theoretical understanding, sample efficiency, scalable coordination, and deployment in real-world settings. Rather than providing an exhaustive systematic review, this survey aims to offer an organised and up-to-date synthesis of recent progress in MARL.

Keywords: multi-agent systems; deep reinforcement learning; cooperative and competitive learning; partial observability; decentralised decision-making

1. Introduction

Multi-agent systems (MAS) arise in many modern applications where multiple autonomous agents interact in a shared environment. For example, autonomous vehicle coordination [1], resource allocation [2], robot swarms [3], UAVs [4], rescue operation [5], and multi-player games [6], among others. In recent years, the combination of deep learning (DL) with reinforcement learning (RL) has produced rapid advances in single-agent decision-making; extending these advances to multi-agent settings, known as multi-agent deep reinforcement learning (MARL). It introduces fundamentally new technical challenges and research directions that have been the subject of extensive study in the literature [7–9].

A well-known characteristic of MARL problems is the non-stationary problem. This is because each agent's policy evolves during learning. From the viewpoint of any single agent, the environment (including other agents) is non-stationary, which complicates convergence analysis and training stability [8]. Other core challenges include partial observability where agents only see local observations, credit assignment across agents



Academic Editor: Keun Ho Ryu

Received: 11 May 2026

Revised: 9 July 2026

Accepted: 4 August 2026

Published: 6 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

for team rewards, and scalability to many agents or large state/action spaces [7,8]. These challenges mean that simply applying single-agent DRL methods often does not work well in multi-agent settings.

To address these challenges, research in MARL has largely focused on a few key paradigms. One of them is *centralised training with decentralised execution* (CTDE). In this paradigm, agents can access additional information during training, such as the global state, other agents' observations, or gradients, to help stabilise learning, while at execution time, each agent operates using only its own local policy [7]. Within CTDE, *value-decomposition* methods split a joint value function into individual agent utilities, which works well for cooperative tasks. Recent studies have improved these methods using distributional and graph-based approaches to make them more powerful and effective [10]. Beyond CTDE and factorisation, recent years have seen several other important directions:

- **Learned communication and differentiable message passing.** Methods that learn when and what agents should communicate, often via learned channels or attention, improve coordination under partial observability. Surveys highlight communication as a major trend and design axis in MARL systems [7].
- **Graph neural networks (GNNs) and relational inductive biases.** GNNs are widely used to model inter-agent interactions explicitly, improving generalisation across different agent counts and topologies [7].
- **Sample efficiency and credit assignment.** Distributional RL, multi-step returns, and improved factorisation schemes have been proposed to accelerate learning and attribute global rewards to local actions [10].
- **Meta-learning, discovery, and automated algorithm search.** Recent work investigates meta-learning and automated discovery of learning rules in multi-agent settings, enabling agents to learn how to adapt across diverse environments and populations. These approaches offer a promising alternative to predefined algorithms by producing more robust and generalisable multi-agent learning strategies [11].

These directions are supported by multiple recent survey and review articles that synthesise empirical trends, theoretical progress, algorithms, and application domains [7–9,11,12]. As the field matures, benchmarks and reproducibility have become increasingly important. Standardised environments and custom simulated domains allow more direct comparison of algorithms as we review in Section 5. However, the survey emphasises concerns about metric consistency, hyperparameter sensitivity, and the need for stronger evaluation protocols to measure generalisation and robustness [8,9].

This paper provides a narrative survey and critical overview of recent developments in DRL for MAS. In particular, we: (1) summarise the central challenges of MARL and formalise the common problem settings underpinning the field; (2) classify recent algorithmic advances, encompassing centralised training with decentralised execution, value decomposition, policy gradient and actor–critic frameworks, model-based approaches, graph-based architectures, and learned communication; (3) review empirical benchmarks and evaluation practices across systemic performance, robustness and adaptability, and domain-specific evaluation; and (4) identify key open problems and promising future directions, including scalable coordination theory, safety and robustness under adversarial conditions, sample efficiency, and real-world deployment constraints. Figure 1 depicts a simple taxonomy of MARL. Following the roadmap shown in Figure 2, the survey first introduces the MARL problem setting and its fundamental challenges, then reviews major algorithmic paradigms, followed by benchmarking and evaluation methodologies. The survey concludes with a discussion of open challenges and future research directions.

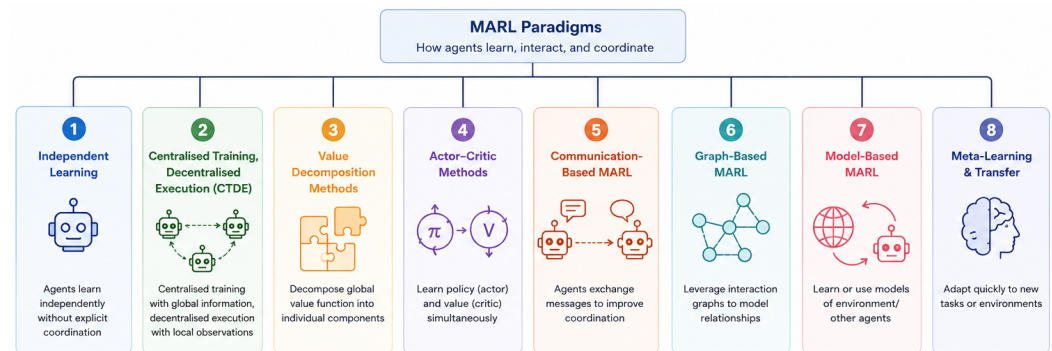


Figure 1. Taxonomy of multi-agent reinforcement learning.

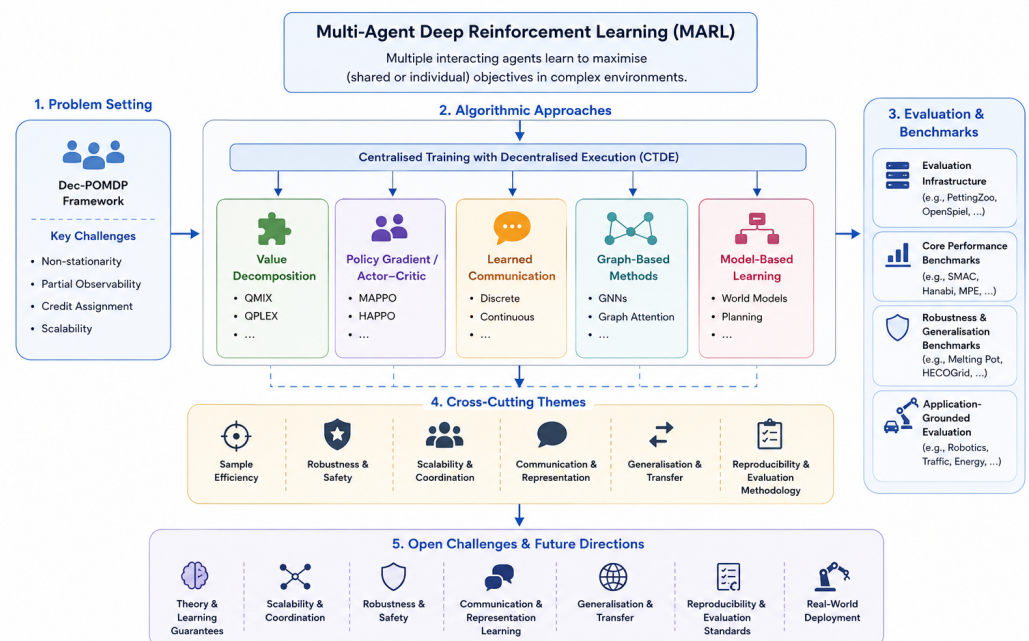


Figure 2. High-level thematic overview and organisation of the survey. The figure illustrates the conceptual structure of the review and the relationships among its major topics.

Paper Organisation

The remainder of the paper is organised as follows: Section 2 presents the background and motivation, concluding with a discussion of previous survey efforts and how the work presented in this paper complements them. Section 3 presents review methodology. Section 4 reviews and analyses DRL techniques and their applications in MAS, including centralised, decentralised, value-based, policy-gradient, model-based, and graph-based methods. Section 5 reviews and analyses common benchmarks and evaluation metrics for assessing MARL algorithms, covering environments, tasks, and literature-reported metrics considering various categories. Section 6 summarises recent advances and outlines open challenges. Finally, Section 7 concludes the paper.

Appendix A overviews multi-agent systems, beginning with their fundamental definition, key characteristics, and capabilities for decentralised problem-solving. The section concludes by highlighting some MAS applications. Appendix B discusses the fundamentals of deep reinforcement learning. Readers without the necessary background may wish to read Appendices A and B before starting Section 2, as they provide a brief overview of the relevant material. Specialist readers may skip Appendices A and B.

2. Background and Motivation

Deep reinforcement learning has emerged as a powerful technique for enabling machines to learn complex decision-making behaviours through continuous interaction with their environment [13]. By integrating reinforcement learning (RL) principles with deep neural networks, DRL overcomes the limitations of classical reinforcement learning methods that rely on tabular representations or manually designed features [14]. This integration allows agents to handle high-dimensional observations, continuous state and action spaces, and nonlinear dynamics, enabling learning directly from raw sensory inputs such as images, signals, or multi-modal data. As a result, DRL has demonstrated significant performance across a wide range of domains, including robotics, autonomous driving, control systems, and strategic gameplay, where explicit modelling of the environment is often infeasible or incomplete [13].

At the same time, many real-world decision-making problems are inherently distributed in nature, which is the central focus of multi-agent systems (MAS) [15]. In a MAS, multiple autonomous agents operate in a shared environment, each pursuing individual or collective goals while interacting with other agents. These interactions may be cooperative, competitive, or mixed, and the actions of one agent can significantly influence the observations, rewards, and future decisions of others. Such settings naturally arise in applications such as robotic swarms, traffic management, distributed energy systems, communication networks, and autonomous vehicle coordination, where decentralised decision-making and coordination among multiple entities are essential [16].

The motivation for combining DRL with MAS arises from the need to enable agents to learn intelligent, adaptive behaviours within the context of complex inter-agent interactions [17]. MARL extends single-agent DRL by allowing multiple agents to simultaneously learn from experience in a shared environment. Through this learning process, agents can acquire coordinated strategies, negotiate resource usage, or develop competitive behaviours without relying on predefined rules or centralised control. This learning-based coordination is particularly useful in dynamic and uncertain environments, where system dynamics may change over time and predefined policies often fail to generalise.

However, integrating DRL into MAS settings introduces additional challenges that are absent in single-agent scenarios. These challenges include non-stationary learning dynamics caused by concurrently updating agent policies, difficulties in assigning credit for global rewards to individual agent actions, partial observability, and scalability to large numbers of agents [17]. Despite these difficulties, the combination of DRL and MAS offers a flexible and powerful framework for building autonomous systems that can learn, cooperate, compete, and evolve in realistic environments. Consequently, MARL has become an active and rapidly growing research area, attracting increasing research attention [18,19].

Existing survey studies (foundational [17,20,21] and more recent [7–9,11,12]) have substantially advanced the understanding of MARL by reviewing its core challenges, methods, algorithms, and application domains. For example, Nguyen et al. [17], Hernandez-Leal et al. [20], and Gronauer and Diepold [21] provide foundational surveys on the challenges of non-stationarity, coordination, and scalability in multi-agent deep reinforcement learning, while more recent works by Zhu et al. [7], Ning and Xie [8], Huh and Mohapatra [9], Oh et al. [11], and Liang et al. [12] extend this discussion to communication mechanisms, contemporary algorithms, and emerging applications. However, most existing surveys either focus on specific dimensions of MARL, such as communication [7], theoretical critique [20], or broad algorithmic categorisation [21], often without sufficiently integrating recent advances in graph-based learning, meta-learning, benchmarking frameworks, and robustness evaluation into a unified critical perspective. The proposed review of this paper differs from these prior surveys by offering a more comprehensive and up-to-date synthesis

that bridges methodological advances with practical deployment challenges, benchmark standardisation, and theoretical limitations. In particular, it moves beyond descriptive summarisation by critically examining issues such as sample efficiency, convergence guarantees, reproducibility, and real-world applicability, therefore addressing key gaps left by earlier survey efforts and providing a roadmap for future research.

3. Review Methodology

3.1. Review Approach

This paper adopts a *narrative survey* methodology rather than a formal systematic review or meta-analysis. The objective is to provide a structured synthesis of major methodological developments, benchmark environments, and emerging research directions in multi-agent deep reinforcement learning (MARL). Although a systematic search strategy was used to identify relevant literature, the review aims to summarise representative and influential contributions rather than exhaustively capture all published studies. Consequently, the categorisation presented in this paper should be interpreted as an organised perspective on the field rather than a complete systematic mapping of MARL research.

3.2. Literature Search Strategy

The literature search was conducted between January and June 2026 using major scientific databases that index research in artificial intelligence, machine learning, robotics, and control systems. The primary databases included IEEE Xplore, ACM Digital Library, Scopus, Web of Science, SpringerLink, ScienceDirect (Elsevier), Google Scholar, and arXiv.

The search focused on publications related to multi-agent reinforcement learning and deep reinforcement learning. Representative search queries included “multi-agent reinforcement learning”, “multi-agent deep reinforcement learning”, “MARL”, “centralised training decentralised execution” OR “CTDE”, “value decomposition networks” OR “QMIX” OR “QPLEX”, “multi-agent communication learning”, “graph neural networks for multi-agent reinforcement learning”, “model-based multi-agent reinforcement learning”, “multi-agent reinforcement learning benchmarks”, “cooperative multi-agent reinforcement learning”, and “competitive multi-agent reinforcement learning”.

Boolean operators and database-specific filtering options were employed where available to improve retrieval precision.

3.3. Time Window

Given the rapid development of MARL, the review primarily focuses on studies published between 2017 and 2026. The year 2017 was selected because it marks the emergence of several influential deep MARL frameworks, including MADDPG, COMA, and subsequent CTDE-based approaches that shaped later research directions. Earlier foundational works on reinforcement learning, multi-agent systems, and decentralised partially observable Markov decision processes (Dec-POMDPs) were included when necessary to provide theoretical context. Also, to provide the necessary theoretical foundation and historical context, seminal technical contributions and influential survey papers published as early as 1992 were also considered where relevant.

3.4. Inclusion and Exclusion Criteria

Studies were considered for inclusion if they satisfied one or more of the following criteria: (i) proposed a novel MARL algorithm, architecture, or training framework; (ii) addressed key challenges such as coordination, communication, credit assignment, scalability, robustness, or sample efficiency; (iii) introduced benchmark environments, evaluation methodologies, or reproducibility frameworks; (iv) reported empirical evaluation on es-

established MARL benchmarks; and (vi) appeared in peer-reviewed journals, conferences, workshops, or highly influential preprints. Exclusion criteria include (i) studies focusing exclusively on single-agent reinforcement learning without multi-agent extensions; (ii) publications lacking methodological or empirical relevance to MARL; (iii) non-English publications; (iv) duplicate records retrieved from multiple databases and (v) editorials, opinion articles, tutorials, or abstracts without substantive technical contributions.

3.5. Screening Procedure

The screening process consisted of three stages: (i) identification—records were collected from the selected databases using the predefined search terms; (ii) title and abstract Screening—publications were reviewed to assess relevance to the scope of MARL; and (iii) full-text assessment—candidate studies were examined in detail to determine their methodological contribution and relevance to the themes covered in this survey.

The search initially produced approximately 850 records. After duplicate removal and title/abstract screening, approximately 320 studies remained. Following full-text assessment, approximately 150–180 publications were retained and used to develop the review. As the paper writing and revision process progressed, the reference list was continuously updated to incorporate newly identified and relevant publications. Because this work follows a narrative survey methodology, these figures are intended to provide transparency regarding the literature selection process rather than constitute a formal PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses)-compliant systematic review.

3.6. Categorisation Framework

To provide a structured overview of the field, the selected literature was grouped into six broad categories reflecting the dominant research directions in MARL. These include centralised training with decentralised execution (CTDE), which addresses non-stationarity through centralised learning and decentralised policies; value decomposition methods, which focus on cooperative credit assignment by factorising joint value functions; communication-based approaches, which learn information-sharing mechanisms to improve coordination under partial observability; graph-based methods, which model inter-agent relationships using graph neural networks and relational representations; model-based approaches, which leverage environment dynamics to improve planning and sample efficiency; and benchmark and evaluation studies, which examine standardised environments, metrics, reproducibility, robustness, and generalisation. Although some algorithms span multiple categories, this taxonomy provides a practical framework for organising and comparing recent developments in MARL.

3.7. Limitations

Several limitations should be acknowledged. First, despite the use of multiple databases and search terms, some relevant studies may not have been captured. Second, many modern MARL algorithms span multiple methodological categories, making classification inherently subjective. Finally, as a narrative survey, this review emphasises representative and influential developments rather than providing an exhaustive or statistically complete account of the entire MARL literature.

4. Deep Reinforcement Learning Approaches for MAS

This section discusses various DRL techniques and their application in MAS, covering centralised training with decentralised execution, independent learning, value-based methods, policy-gradient methods, model-based methods, multi-agent actor–critic algorithms, learned communication, and graph-based architectures. Advantages, applications, and

open challenges arising from each family are synthesised in the subsequent Section 6. A consolidated comparison of the families—across training and execution requirements, reward setting, scalability, and principal strengths and limitations—is provided in Table 1 at the end of the section.

4.1. Centralised Training with Decentralised Execution (CTDE)

The paradigm of centralised training with decentralised execution (CTDE) has emerged as a foundational framework in cooperative MARL, addressing fundamental challenges that arise when multiple agents must learn to coordinate in partially observable environments. Figure 3 depicts a MARL framework with CTDE.

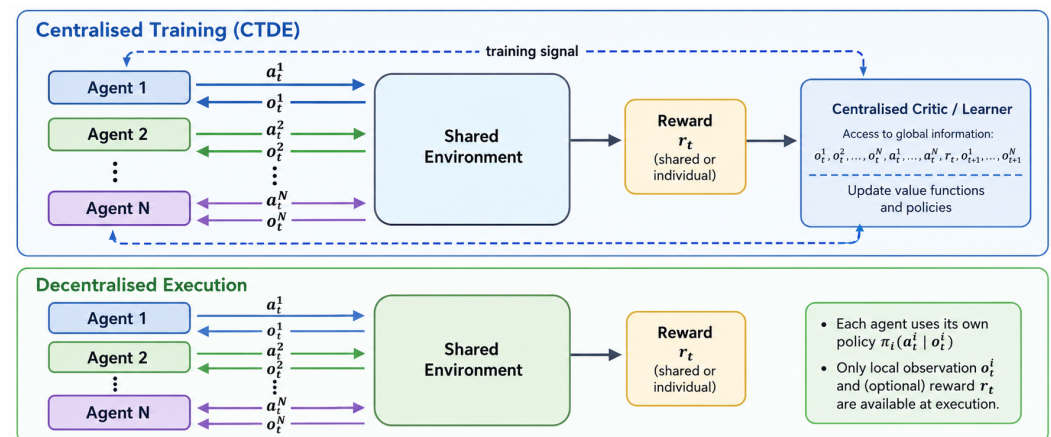


Figure 3. Multi-agent reinforcement learning framework with CTDE.

The core premise of CTDE is deceptively elegant: during training, agents can leverage privileged information—including the global state of the environment, the observations and actions of other agents, and even the underlying dynamics—to learn more effectively. However, during deployment (execution), each agent must operate using only its local observations, adhering to the constraints of real-world scenarios where communication may be limited, delayed, or entirely infeasible [22,23].

This asymmetry between training and execution represents a pragmatic reconciliation of two competing imperatives. On the one hand, decentralised execution is often a strict requirement imposed by physical, computational, or communication constraints in deployed systems. On the other hand, centralised training conditions the learning signal on global information to *mitigate*—rather than eliminate—the non-stationarity intrinsic to MARL, wherein each agent’s learning process is confounded by the concurrent adaptation of other agents’ policies [16,20]. It is important to stress that this mitigation is partial: non-stationarity persists through simultaneous policy updates, replay-buffer staleness, residual partial observability, changing agent populations, learned communication, and critic misspecification, none of which a centralised critic removes by construction. By furnishing a richer and more stable training signal, however, CTDE makes it substantially easier to discover coordinated behaviours that fully decentralised learning struggles to attain.

4.1.1. Mathematical Formulation and Problem Setting

To formalise the CTDE setting, we adopt the standard decentralised partially observable Markov decision process (Dec-POMDP) [22], defined by the tuple

$$\langle \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \Omega, \{O^i\}_{i \in \mathcal{N}}, r, \gamma, \rho_0 \rangle,$$

where $\mathcal{N} = \{1, \dots, n\}$ is the set of n agents and $\mathcal{S}$ is the global state space. The *joint action space* is $\mathcal{A} = \prod_{i \in \mathcal{N}} \mathcal{A}^i$, with $\mathcal{A}^i$ the action space of agent i and a joint action written $\mathbf{a} = (a^1, \dots, a^n) \in \mathcal{A}$; the transition kernel is $\mathcal{P}(s' \mid s, \mathbf{a})$. The *joint observation space* is $\Omega = \prod_{i \in \mathcal{N}} \Omega^i$, with $\mathbf{o} = (o^1, \dots, o^n)$ and per-agent observation function $O^i : \mathcal{S} \rightarrow \Omega^i$ (which generalises to a stochastic emission $O(\mathbf{o} \mid s, \mathbf{a})$). Finally, $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the shared team reward, $\gamma \in [0, 1)$ is the discount factor, and ρ_0 is the initial state distribution. Under partial observability, agent i acts on its local action–observation history $\tau_t^i = (o_0^i, a_0^i, \dots, a_{t-1}^i, o_t^i) \in \mathcal{T}^i$, and its policy is a conditional distribution $\pi_{\theta_i}^i(a^i \mid \tau^i)$ parameterised by θ_i (reducing to the memoryless form $\pi_{\theta_i}^i(a^i \mid o^i)$ when histories are not retained). Writing the joint policy as $\pi_{\theta} = (\pi_{\theta_1}^1, \dots, \pi_{\theta_n}^n)$, the cooperative objective is to maximise the expected discounted return

$$J(\theta) = \mathbb{E}_{s_0 \sim \rho_0, \mathbf{a}_t \sim \pi_{\theta}, s_{t+1} \sim \mathcal{P}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \mathbf{a}_t) \right]. \quad (1)$$

4.1.2. The Centralised Training Paradigm

During training, CTDE methods exploit privileged information beyond any single agent’s observational capacity: a centralised critic or value function may condition on the global state s_t , the joint action $\mathbf{a}_t$, or the joint history, furnishing a lower-variance and more stationary target than a purely local signal [24,25]. Two principal instantiations of this idea recur throughout the cooperative MARL literature. The first conditions a *centralised critic* on global information whilst retaining decentralised actors; the canonical example, MADDPG [24], together with its attention-based and factorised successors, is examined in detail in Section 4.6. The second *factorises* the joint action–value function into per-agent utilities depending only on local information, enabling decentralised greedy execution; this value-decomposition family (VDN, QMIX, QTRAN, QPLEX) is treated in Section 4.3.

A distinctive CTDE mechanism for the multi-agent credit-assignment problem is the counterfactual baseline of COMA [25], which uses the centralised critic to compute, for each agent i , the advantage

$$A^i(s, \mathbf{a}) = Q(s, \mathbf{a}) - \sum_{a^i \in \mathcal{A}^i} \pi^i(a^i \mid \tau^i) Q(s, (\mathbf{a}^{-i}, a^i)), \quad (2)$$

marginalising over agent i ’s own actions whilst holding the other agents’ actions $\mathbf{a}^{-i}$ fixed. This isolates the marginal contribution of agent i to the team return, yielding a low-variance, agent-specific learning signal without requiring a separately designed per-agent reward.

4.1.3. The Decentralised Execution Paradigm

At execution time, each agent acts using only its local information: its policy conditions on the current observation o_t^i or, under partial observability, on the local action–observation history τ_t^i , with no access to the global state or to other agents’ observations and actions. To summarise long histories compactly, recurrent encoders such as Long Short-Term Memory (LSTM) [26] or Gated Recurrent Unit (GRU) [27] cells maintain a latent state $h_t^i = f_{\text{rec}}(h_{t-1}^i, o_t^i, a_{t-1}^i)$, so that the executable policy takes the form $\pi_{\theta_i}^i(a_t^i \mid h_t^i)$ [28]. Where limited inter-agent communication is permitted at execution time, the strict locality of CTDE can be relaxed to *partially* decentralised execution; learned communication protocols of this kind are discussed in Section 4.7.

4.2. Independent Learning Approaches

Independent learning (IL) represents the conceptually simplest approach to multi-agent reinforcement learning, wherein each agent applies a standard single-agent RL

algorithm whilst treating all other agents as components of a non-stationary environment. No explicit coordination mechanism, shared value function, centralised critic, or inter-agent communication is assumed at any stage. The seminal distinction between independent learners and joint action learners was formalised by Claus and Boutilier [29], who established that although agents lacking knowledge of one another's actions suffer an informational disadvantage, they need not necessarily converge to worse outcomes than agents explicitly tracking the joint action space in fully cooperative, repeated games.

The canonical instantiation, Independent Q-Learning (IQL), proposed by Tan [30], applies the standard temporal-difference update independently for each agent i :

$$Q^i(o^i, a^i) \leftarrow Q^i(o^i, a^i) + \alpha \left[r^i + \gamma \max_{a^{ii}} Q^i(o^{ii}, a^{ii}) - Q^i(o^i, a^i) \right], \quad (3)$$

where o^i denotes the local observation of agent i , α is the learning rate, and r^i is the scalar reward received. Because each agent's Q-function operates over only its local observation-action space $\Omega^i \times \mathcal{A}^i$, IQL avoids the exponential growth of the joint action space $\prod_{i \in \mathcal{N}} \mathcal{A}^i$ that provides centralised value methods intractable at scale. The extension to deep function approximators—yielding Independent Deep Q-Networks (IDQN) [31]—preserves this scalability whilst enabling learning from high-dimensional perceptual inputs.

4.2.1. Theoretical Limitations and Coordination Failures

The fundamental theoretical limitation of IL is non-stationarity. As each agent's policy π^i evolves concurrently during training, the effective transition function perceived by any single agent is a time-varying function of the joint policy π^{-i} , violating the stationarity assumption underpinning convergence guarantees and invalidating the conditions of Watkins and Dayan's convergence proof [32,33]. Matignon et al. [34] identified five distinct coordination failure modes: (i) *Pareto-selection*; (ii) *non-stationarity*; (iii) *stochasticity*, wherein reward variance induced by simultaneously exploring teammates produces contradictory training signals; (iv) *alter-exploration*, wherein an agent's exploratory deviations are misinterpreted as genuine policy shifts; and (v) *shadowed equilibria*, in which the optimal joint equilibrium is hidden from individual agents' perspectives. This taxonomy is significant because non-stationarity, whilst the most widely discussed challenge, is only one of several distinct and partially independent failure modes.

4.2.2. Mitigation Strategies

Several algorithmic modifications mitigate the failure modes of IL. Hysteretic Q-learning [35] introduces an asymmetric learning rate schedule: a larger rate α is applied to positive TD errors, whilst a smaller rate $\beta < \alpha$ governs negative ones, preventing premature policy pessimism caused by exploratory actions of teammates. Lenient learners [34] extend this by averaging only over the highest rewards per state-action pair, reducing sensitivity to teammate-induced stochasticity. Win-or-Learn-Fast (WoLF) adjusts the learning rate dynamically based on whether the current policy performs above or below the expected Nash equilibrium payoff. Despite these extensions, all such methods constitute post hoc corrections that reduce the severity of non-stationarity without resolving it, and convergence guarantees remain restricted to simplified settings [34]. Table 1 situates independent learning against coordinated alternatives, deliberately separating its low computational cost and minimal information requirements (a practical advantage) from its weak convergence guarantees and susceptibility to coordination failure (a theoretical liability). The practical competitiveness and scalability implications of IL are discussed further in Section 6.

4.3. Value-Based Methods

Value-based reinforcement learning methods learn an explicit approximation of the optimal action-value function $Q^*(s, a)$ and derive a greedy policy implicitly via $\pi^*(s) = \arg \max_a Q^*(s, a)$. The theoretical foundation is the Q-learning algorithm of Watkins and Dayan [32], which converges to Q^* in the tabular setting. The Bellman optimality equation is:

$$Q^*(s, a) = \mathbb{E}_{s'} \left[r(s, a) + \gamma \max_{a'} Q^*(s', a') \right]. \quad (4)$$

Deep Q-Networks (DQN) [31] extended this to high-dimensional perceptual domains via a deep convolutional network parameterised by θ , stabilised by experience replay and a periodically frozen target network θ^- :

$$\mathcal{L}(\theta) = \mathbb{E}_{(s, a, r, s') \sim \mathcal{D}} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right]. \quad (5)$$

DQN achieved human-level performance on 49 Atari 2600 games from raw pixels, demonstrating the viability of end-to-end value-based learning. However, the architecture was accompanied by several well-documented failure modes that motivated a subsequent generation of algorithmic refinements.

4.3.1. Algorithmic Refinements and Critical Analysis

A fundamental issue with DQN is overestimation bias: the max operator over noisy estimates simultaneously selects and evaluates the greedy action using the same network. Van Hasselt, Guez, and Silver [36] demonstrated this produces systematic overestimation across the Atari benchmark and proposed Double DQN to decouple action selection from evaluation:

$$y_t = r + \gamma Q \left(s', \arg \max_{a'} Q(s', a'; \theta); \theta^- \right). \quad (6)$$

The Duelling Network Architecture [37] further decomposes the Q-function into a state-value stream $V(s; \theta, \beta)$ and an advantage stream $A(s, a; \theta, \alpha)$:

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + \left(A(s, a; \theta, \alpha) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(s, a'; \theta, \alpha) \right), \quad (7)$$

enabling accurate state valuations independently of per-action advantage estimates. Prioritised Experience Replay [38] improves sample efficiency by sampling transitions proportional to their TD-error magnitude. The Rainbow framework [39] integrated six complementary improvements—Double DQN, duelling networks, Prioritised Replay, multi-step returns, distributional RL, and Noisy Networks—with ablation analysis confirming that distributional RL and Prioritised Replay contributed the largest individual gains, whilst the contribution of Noisy Networks was task-dependent.

The distributional perspective [40] models the full return distribution $Z^\pi(s, a)$ via the distributional Bellman equation:

$$Z^\pi(s, a) \stackrel{D}{=} R(s, a) + \gamma Z^\pi(s', A'), \quad (8)$$

where $\stackrel{D}{=}$ denotes equality in distribution. The categorical algorithm C51 parameterises this distribution over 51 fixed atoms. The subsequent QR-DQN [41] removes the fixed-support constraint via a quantile function parameterisation, providing a theoretically consistent contraction in the 1-Wasserstein metric and yielding further performance improvements. A broader theoretical concern is the *deadly triad* [42]: the combination of function approxi-

mation, bootstrapping, and off-policy learning, which van Hasselt et al. [43] empirically characterised as a frequent source of divergence in deep value-based learning. DQN manages this instability through target networks and replay buffers, but these are engineering mitigations rather than theoretical solutions.

4.3.2. Value Decomposition in Multi-Agent Settings

In cooperative MARL, the joint action-value function $Q_{\text{tot}}(\mathbf{s}, \mathbf{a})$ grows exponentially in agent count, necessitating factorisation. The Individual-Global-Max (IGM) condition [44] characterises the requirement for consistency between centralised and decentralised greedy action selection:

$$\arg \max_{\mathbf{a}} Q_{\text{tot}}(\mathbf{s}, \mathbf{a}) = \left(\arg \max_{a^1} Q^1(o^1, a^1), \dots, \arg \max_{a^n} Q^n(o^n, a^n) \right). \quad (9)$$

Intuitively, IGM guarantees that the joint action maximising the centralised Q_{tot} can be recovered by each agent independently maximising its own local utility Q^i , so that decentralised greedy execution remains consistent with the centrally trained joint value. VDN [45] enforces IGM through additive decomposition, and QMIX [46] relaxes additivity to a monotonic mixing network ($\frac{\partial Q_{\text{tot}}}{\partial Q^i} \geq 0$), which is sufficient for IGM because a positive-weighted combination preserves each agent's per-utility arg max. The price is representational: monotone mixers cannot express tasks whose optimal joint value is non-monotone in the individual utilities. Wang et al. [47] demonstrated through formal analysis that this monotonicity constraint imposes non-trivial representational limitations: the class of joint value functions expressible by QMIX is strictly smaller than the full IGM-compliant class, and there exist cooperative tasks requiring non-monotonic joint strategies for which QMIX provably fails to represent the optimal joint value function. QTRAN [44] addresses this through a more flexible optimisation objective, but the resulting procedure is computationally demanding and unstable in high-dimensional environments [48]. QPLEX [48] achieves full IGM expressiveness via a duplex duelling architecture that decomposes the joint advantage function, providing QTRAN's representational benefits with greater training stability.

4.4. Policy Gradient Methods

Policy gradient methods directly optimise a parameterised stochastic policy $\pi_{\theta}(a | s)$ with respect to $J(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta}}[\sum_{t=0}^{\infty} \gamma^t r_t]$ by ascending its gradient. The policy gradient theorem [49] establishes the tractable form:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t=0}^T \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot G_t \right], \quad (10)$$

forming the basis of REINFORCE [50]. Variance is reduced without bias by subtracting the state-value baseline $V^{\pi}(s_t)$:

$$\nabla_{\theta} J(\theta) \approx \mathbb{E}_t [\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot \hat{A}_t], \quad \hat{A}_t = G_t - V^{\pi}(s_t). \quad (11)$$

4.4.1. Actor–Critic Methods and Trust-Region Algorithms

The actor–critic architecture substitutes Monte Carlo return estimates with bootstrapped critic estimates, with the generalised advantage estimator (GAE) [51] interpolating the variance–bias trade-off via eligibility trace parameter $\lambda \in [0, 1]$. Asynchronous Advantage Actor–Critic (A3C) [52] addressed sample inefficiency through parallelised actors asynchronously updating a shared parameter server, though asynchronous gradient staleness can degrade performance relative to synchronous alternatives. Trust Region Policy

Optimisation (TRPO) [51] formalised stable updates by constraining successive policies within a KL-divergence ball:

$$\max_{\theta} \mathbb{E}_{s,a \sim \pi_{\theta_{\text{old}}}} \left[\frac{\pi_{\theta}(a | s)}{\pi_{\theta_{\text{old}}}(a | s)} \hat{A}^{\pi_{\theta_{\text{old}}}}(s, a) \right] \quad \text{s.t.} \quad \mathbb{E}_s [D_{\text{KL}}(\pi_{\theta_{\text{old}}} \| \pi_{\theta})(s)] \leq \delta. \quad (12)$$

Proximal Policy Optimisation (PPO) [53] replaced the constrained optimisation of TRPO with a clipped surrogate:

$$\mathcal{L}^{\text{CLIP}}(\theta) = \mathbb{E}_t [\min(\rho_t(\theta) \hat{A}_t, \text{clip}(\rho_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t)], \quad (13)$$

where $\rho_t(\theta) = \pi_{\theta}(a_t | s_t) / \pi_{\theta_{\text{old}}}(a_t | s_t)$. PPO's simplicity and competitive performance have made it the default policy gradient baseline in both single-agent and multi-agent research. In cooperative MARL, Multi-Agent PPO (MAPPO) [54] applies per-agent PPO updates within the CTDE framework, pairing independent actors with a centralised value function conditioned on the global state.

4.4.2. Critical Analysis

A significant critique of PPO's reputation was advanced by Engstrom et al. [55], who demonstrated that performance improvements attributed to PPO over TRPO are largely attributable to undocumented *code-level optimisations*—including reward normalisation, value function clipping, gradient norm clipping, and orthogonal weight initialisation—rather than to the clipped surrogate objective itself. Ablation studies showed that PPO with these implementation details removed performs comparably to TRPO. This finding carries substantial implications for algorithmic attribution in deep RL, reinforcing the need for controlled comparative evaluations that isolate each component's contribution independently. In a carefully controlled benchmarking study across SMAC and multi-agent MuJoCo, Yu et al. [54] reported empirically that MAPPO achieves performance competitive with—and in several scenarios superior to—QMIX and other off-policy value decomposition methods, challenging the prevailing assumption that on-policy policy gradient methods are inherently sample-inefficient in cooperative MARL. This finding contributed to a broader revisionary discussion [56] in which performance rankings of MARL algorithms were shown to be sensitive to hyperparameter tuning, parameter sharing configurations, and task structure.

4.5. Model-Based Methods

Model-based reinforcement learning (MBRL) augments the agent's learning loop with an explicit predictive model of transition dynamics $\hat{T} : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ and reward function $\hat{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, learned from observed transitions. Whilst models improve *sample efficiency* relative to model-free methods, model inaccuracies introduce *model bias*: a systematically distorted training signal that can lead to policies performing well in the model but poorly in the true environment. Moerland et al. [57] provide a comprehensive taxonomy of MBRL approaches, distinguishing methods based on dynamics model representation, planning procedure, and the degree of coupling between model learning and policy optimisation.

4.5.1. Key Algorithms and Their Limitations

The Dyna architecture [58] introduced the foundational principle of interleaving real transitions with model-generated simulated experience, performing k additional Q-learning updates per real interaction. In deep function approximator settings, errors accumulate over multi-step rollouts: for a dynamics model with per-step error ϵ_m , trajectory deviation over horizon H scales as $O(H\epsilon_m)$ [59], providing long-horizon rollouts unreliable even for moderately accurate models. Model Predictive Control (MPC) mitigates this by re-planning

at each decision step. PETS [60] demonstrated that an ensemble of probabilistic neural network dynamics models enables principled uncertainty-aware planning, substantially outperforming model-free baselines in sample efficiency on continuous locomotion tasks. Monte Carlo Tree Search (MCTS) [61], integrated with deep neural network priors in AlphaZero [62] and extended to latent-space planning in MuZero [63], enables superhuman performance in combinatorially rich domains at the cost of high per-step inference time. The Model-Based Policy Optimisation (MBPO) algorithm [59] demonstrated that *short* model-generated rollouts ($k \leq 10$ steps) branched from real data achieve one to two orders of magnitude greater sample efficiency than competitive model-free methods on MuJoCo benchmarks, without incurring excessive compounding errors. Dreamer [64] learns a recurrent world model in a compact latent space and trains both actor and critic through backpropagation through imagined latent rollouts; a critical limitation is that the world model must accurately capture global task-relevant dynamics from observations alone, which may fail under partial observability or stochasticity, leading to systematic value overestimation in imagined trajectories.

4.5.2. Multi-Agent Dynamics Modelling: Challenges

In multi-agent settings, the joint transition kernel $\mathcal{P}(s_{t+1} \mid s_t, \mathbf{a}_t)$ depends on the simultaneous actions of all agents, introducing structural complexity absent from single-agent MBRL along three axes. First, *model representation*: one may learn either a centralised dynamics model of the full joint system [16], which is accurate but suffers the curse of dimensionality as the joint action space $\prod_{i \in \mathcal{N}} \mathcal{A}^i$ grows, or agent-conditioned factorised models that explicitly represent the influence of other agents—a form of opponent modelling that trades dimensionality for additional estimation error [20]. Second, *planning with joint actions*: optimising or sampling rollouts over the exponentially large joint action space is far more expensive than in the single-agent case, so practical methods typically restrict planning to short horizons or to factorised action representations. Third, *model bias under co-adaptation*: the multi-agent analogue of compounding error is especially severe because the very quantity being modelled is non-stationary—even a model that is accurate at initialisation degrades as agent policies co-evolve, creating a self-reinforcing feedback loop between model error and policy drift. Consequently, learned multi-agent world models remain most reliable when interactions are sparse and coordination structures evolve slowly; their scalability to large populations under tight real-time budgets is largely unestablished, a point taken up further in Section 6.

4.6. Multi-Agent Actor–Critic Algorithms

Multi-agent actor–critic (MAAC) algorithms combine policy gradient optimisation with value-function critics providing low-variance learning signals. Each agent i maintains an actor $\pi_{\theta_i}^i$ mapping local observations to distributions over actions, whilst a centralised critic estimates state or state–action values during training. The Deterministic Policy Gradient (DPG) theorem of Silver et al. [65] provided the theoretical foundation for continuous action spaces, showing that the gradient of the expected return with respect to a deterministic policy $\mu_{\theta} : \mathcal{S} \rightarrow \mathcal{A}$ takes the form:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{s \sim \rho^{\mu}} \left[\nabla_{\theta} \mu_{\theta}(s) \cdot \nabla_a Q^{\mu}(s, a) \Big|_{a=\mu_{\theta}(s)} \right], \quad (14)$$

enabling efficient off-policy gradient estimation without variance-inducing stochasticity. This result underpins both DDPG [66] and MADDPG [24].

4.6.1. MADDPG: Contribution and Critique

MADDPG equips each agent i with a centralised action-value function $Q_{\phi_i}^i(s, a^1, \dots, a^n)$ conditioned on the global state and joint actions, with actor updates:

$$\mathcal{L}(\phi_i) = \mathbb{E} \left[\left(Q_{\phi_i}^i(s, \mathbf{a}) - (r_i + \gamma Q_{\phi_i}^i(s', \mathbf{a}')) \right)^2 \right], \quad (15)$$

$$\nabla_{\theta_i} J(\theta_i) = \mathbb{E} \left[\nabla_{\theta_i} \pi_{\theta_i}^i(o^i) \cdot \nabla_{a^i} Q_{\phi_i}^i(s, a^1, \dots, a^n) \Big|_{a^i = \pi_{\theta_i}^i(o^i)} \right]. \quad (16)$$

MADDPG demonstrated effective policy learning in mixed cooperative–competitive particle environments. However, maintaining n independent centralised critics each conditioning on the full joint action tuple scales the total critic parameter count as $O(n^2)$, a prohibitive burden in large agent populations [21]. The algorithm further inherits DDPG’s known instabilities [66], including sensitivity to hyperparameter settings and susceptibility to overestimation bias; when compared directly to MAPPO in controlled experiments [54], MADDPG was outperformed on several SMAC scenarios and multi-agent MuJoCo benchmarks.

4.6.2. Attention-Based and Factorised Actor–Critic Architectures

The Multi-Actor-Attention-Critic (MAAC) algorithm [67] addressed MADDPG’s scalability limitation by replacing per-agent joint critics with a multi-head self-attention mechanism:

$$V^i = f^i \left(o^i, a^i, \sum_{j \neq i} \alpha_{ij} h^j \right), \quad (17)$$

where $h^j = g(o^j, a^j)$ encodes agent j ’s local information, α_{ij} is a learned attention weight, and f^i is a mixing function. This reduces effective input complexity from $O(n)$ to the size of the aggregated context vector. Nevertheless, MAAC provides no theoretical guarantees on the faithfulness of attention aggregation—the mechanism may discard relevant coordination information if its inductive biases do not match the cooperative task structure. FACMAC [68] addressed the representational limitations of independent critics by factorising the centralised Q-function analogously to QMIX and applying policy gradients through the factorised critic jointly across all agents’ actors, capturing inter-agent coordination signals invisible to per-agent critic methods. Empirical evaluations demonstrated significant advantages over MADDPG on SMAC and multi-agent MuJoCo tasks, particularly at higher agent counts. The HAPPO algorithm [69] established the first formal monotonic joint-policy improvement guarantee for heterogeneous cooperative MARL through a sequential trust-region update scheme wherein each agent’s update uses an importance sampling correction accounting for preceding agents’ updated policies. A practical limitation is that the per-update computational cost scales linearly in agent count, and update ordering may affect convergence speed in large populations.

4.6.3. Critical Synthesis

Gronauer and Diepold [21] identify three persistent challenges insufficiently addressed by current MAAC approaches. First, training instability caused by simultaneous actor and critic updates across multiple agents remains a practical obstacle, with gradient interference producing non-monotonic learning curves and high sensitivity to initialisation. Second, theoretical grounding is incomplete: whilst HAPPO provides monotonic improvement guarantees, MADDPG, MAAC, and FACMAC rely primarily on empirical validation without convergence analysis. Third, the scalability of centralised critics to real-world problems—involving hundreds of agents, continuous state spaces, and real-time

constraints—has not been demonstrated convincingly, with most benchmarks involving fewer than 30 agents in simulation.

4.7. Communication-Based Methods

Under partial observability, agents can overcome the limits of purely local information by *learning to communicate*: exchanging differentiable messages whose content and timing are optimised jointly with the policy rather than hand-designed. The seminal differentiable inter-agent learning (DIAL) and reinforced inter-agent learning (RIAL) approaches of Foerster et al. [23] demonstrated that, by allowing gradients to flow through a communication channel during centralised training, agents can discover compact protocols that are subsequently executed with discretised messages. CommNet [70] replaces an explicit channel with a continuous broadcast in which each agent’s hidden state is averaged and fed back as a communication vector, enabling end-to-end learning of cooperative behaviour but providing no mechanism to address messages selectively. TarMAC [71] introduces *targeted* communication through a signature–query attention mechanism: each sender emits a key alongside its message and each receiver weights incoming messages by relevance, allowing many-to-many communication that scales more gracefully with agent count and remains interpretable.

Communication-based methods differ principally along four design axes: *what* is communicated (raw observations, learned embeddings, or intentions), *when* communication occurs (every step, gated, or event-triggered), the *differentiability* of the channel (fully differentiable channels trained by backpropagation versus discrete messages trained by reinforcement), and the *bandwidth* assumed at execution. A recurring caveat is that any method requiring message exchange at execution time relaxes the strict decentralisation of CTDE and presupposes communication infrastructure of adequate reliability and bandwidth—an assumption that may not hold in the deployment settings discussed in Section 6.

4.8. Graph-Based Approaches for Multi-Agent Systems

The algorithmic families surveyed in preceding subsections share a representational assumption: that the multi-agent system can be adequately described by concatenating individual agent observations into a global state vector, or by conditioning value functions on fixed-size joint inputs. This assumption becomes progressively untenable as system scale grows. First, in physically situated MAS—including robot swarms, vehicular networks, and distributed sensor arrays—agents interact primarily with a local neighbourhood whose topology carries task-relevant relational information that flat vector representations discard. Second, methods conditioning on fixed-size joint inputs cannot generalise across agent populations of varying size. Graph Neural Networks (GNNs) [72–74] offer a natural framework for encoding relational structure explicitly, and can exploit sparse interaction topologies and—under suitable conditions on architecture and task topology—generalise across population sizes [7,75].

4.8.1. Graph-Theoretic Formalisation of Multi-Agent Interaction

The multi-agent system at time t is represented as a graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$, where node $v_i \in \mathcal{V}$ corresponds to agent i and edge $(i, j) \in \mathcal{E}_t$ encodes an interaction channel between agents i and j . GNNs operate through iterative message passing [75]:

$$\mathbf{h}^{i,(k+1)} = \phi^{(k)}\left(\mathbf{h}^{i,(k)}, \bigoplus_{j \in \mathcal{N}(i)} \psi^{(k)}(\mathbf{h}^{i,(k)}, \mathbf{h}^{j,(k)}, \mathbf{e}_{ij})\right), \quad (18)$$

whose symbols are defined in Table 2. After K rounds, the node representation $\mathbf{h}^{i,(K)}$ encodes a K -hop neighbourhood context used as input to the agent's value function or policy network. This framework subsumes GCNs [73]—in which messages are computed by a shared linear transform normalised by node degree—and GATs [74]—in which messages are weighted by learned attention coefficients.

4.8.2. Graph Convolutional Reinforcement Learning

The Deep Graph Network (DGN) of Jiang et al. [76] integrated graph convolution directly into the MARL value estimation pipeline by constructing a dynamic k -nearest-neighbour graph and applying multi-head dot-product attention within each convolutional layer. A *relation kernel* computes soft attention coefficients from node feature pairs, providing a differentiable approximation to the graph structure. Agent representations are fed to individual Q-networks under a parameter-sharing scheme enabled by the permutation equivariance of the GNN architecture, allowing the same network weights to service agents in different graph positions. Empirical evaluations on Pommerman, Jungle, and large-scale battle environments demonstrated that DGN substantially outperforms QMIX [46] and IDQN as agent count scales.

4.8.3. Coordination Graphs and Deep Coordination Graphs

Coordination graphs [77] represent agent dependencies as an undirected graph in which each edge $(i, j) \in \mathcal{E}$ is associated with a pairwise payoff $f_{ij}(a^i, a^j)$, yielding the decomposition:

$$Q_{\text{tot}}(s, \mathbf{a}) = \sum_{i \in \mathcal{V}} f_i(a^i) + \sum_{(i,j) \in \mathcal{E}} f_{ij}(a^i, a^j), \quad (19)$$

with the optimal joint action computable in polynomial time for graphs of bounded treewidth via variable elimination or max-sum belief propagation. Deep Coordination Graphs (DCG) [78] extended this to deep function approximation within the CTDE framework, parameterising individual and pairwise payoffs as neural networks and computing the optimal joint action via differentiable max-sum message passing. DCG's representational class subsumes both VDN and QMIX: the pairwise decomposition captures inter-agent dependencies that purely additive factorisations cannot represent, placing DCG on the expressiveness–tractability frontier between additive factorisation and full centralised estimation.

4.8.4. Graph Attention Mechanisms and Dynamic Graph Construction

GATs [74] replace fixed aggregation weights with a learnable attention mechanism computing scalar coefficients between adjacent nodes:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W}\mathbf{h}^i \parallel \mathbf{W}\mathbf{h}^j]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W}\mathbf{h}^i \parallel \mathbf{W}\mathbf{h}^k]))}, \quad (20)$$

endowing each agent with the capacity to selectively weight neighbour contributions. Liu et al. [79] proposed G2ANet employing a two-stage mechanism: *hard attention* produces a binary adjacency matrix by sampling discrete edge-presence decisions, pruning irrelevant interactions; *soft attention* then assigns continuous weights to retained edges. This sparsification reduces communication overhead and over-aggregation artefacts whilst producing interpretable task-relevant interaction structures.

Table 1. Comparison of multi-agent DRL method families across training and execution requirements, reward setting, scalability, and principal strengths and limitations. Representative methods are listed for each family. Citations indicate the primary source for each representative method; the final column is additionally referenced to the surveys and analyses that substantiate the stated limitations.

Family (Representative Methods)	Training/Execution	Reward Setting	Scalability	Key Strengths	Key Limitations
Independent learning (IQL [30], IDQN [31])	Decentralised training; local execution	Any (shared or individual)	Excellent; constant per-agent cost	Trivial to implement; needs no global state or communication	Non-stationarity; coordination failures; weak convergence guarantees [34]
Value decomposition (VDN [45], QMIX [46], QTRAN [44], QPLEX [48])	CTDE with centralised mixer; local greedy execution	Cooperative (shared)	Good; factorised critic	Principled cooperative credit assignment; IGM-consistent decentralised execution	Additivity/monotonicity limit expressiveness; cooperative-only [47]
Centralised-critic actor-critic (MADDPG [24], COMA [25], MAPPO [54], HAPPO [69], FACMAC [68])	CTDE with centralised critic; decentralised actors	Cooperative, competitive, or mixed	Moderate; critic input grows with agents unless factorised or attentive	Continuous actions; mixed-motive settings; lower-variance updates	Training instability; limited convergence theory; critic scalability [21]
Communication-based (RIAL/DIAL [23], CommNet [70], TarMAC [71])	CTDE with learned messages; execution-time message exchange	Mostly cooperative	Depends on bandwidth and topology	Mitigates partial observability; learns what and when to share	Requires execution-time communication; bandwidth assumptions [7]
Graph-based (DGN [76], DCG [78], DICG [80])	CTDE over an interaction graph; local K-hop message passing	Cooperative or mixed	Strong; locality and permutation equivariance enable transfer across sizes	Relational inductive bias; generalisation across agent counts	Sensitive to graph construction; over-smoothing; attention cost [7]
Model-based (Dyna [58], MBPO [59], Dreamer [64]; multi-agent extensions [16, 20])	Learn dynamics and plan/imagine; decentralised actors	Any	Limited; joint-dynamics dimensionality	High sample efficiency in low-interaction regimes	Model bias amplified by co-adaptation; costly joint-action planning [57]

4.8.5. Implicit Graph Structure Learning

Deep Implicit Coordination Graphs (DICG) [80] learn a latent interaction graph directly from agent observations by computing a soft attention matrix $\mathbf{A} \in [0, 1]^{n \times n}$, where entry A_{ij} reflects the inferred dependency between agents i and j , used as the dynamic adjacency for GNN-based value factorisation jointly optimised with the CTDE training objective. DICG was evaluated on cooperative navigation, predator–prey, and StarCraft II scenarios, demonstrating competitive performance with explicitly constructed graph methods in tasks where the relevant interaction topology is non-obvious. Learned attention matrices

consistently exhibit sparsity and structured patterns correlating with ground-truth task dependencies. These advantages are, however, contingent: performance depends on the quality of the (often learned) graph construction, deep message passing can induce over-smoothing that homogenises agent embeddings, attention-based aggregation adds computational cost, and inferring a useful interaction graph under partial observability is itself difficult. The structural advantages, open representational questions, and scalability considerations arising from graph-based approaches are discussed further in Section 6.

4.9. Comparative Summary of Method Families

Table 1 consolidates the families surveyed above, contrasting their training and execution requirements, the reward settings they support, their scalability characteristics, and their main strengths and limitations.

Table 1 makes explicit a recurring trade-off: methods that demand the least coordination machinery (independent learning) scale most readily but offer the weakest guarantees, whereas methods that exploit centralised information or relational structure (value decomposition, centralised-critic actor–critic, and graph-based methods) coordinate more effectively at the cost of additional training-time complexity and, in several cases, restrictive representational assumptions. No single family dominates across reward settings, agent counts, and observability regimes; the appropriate choice is therefore governed by the dominant challenge of the target task—credit assignment, partial observability, mixed-motive interaction, or sample efficiency—rather than by any universal performance ranking. Empirical rankings reported in the literature should accordingly be read as benchmark- and implementation-specific rather than as general statements of superiority [55,56].

Table 2. Notation for the graph-based multi-agent formulation.

Symbol	Meaning
$\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$	Interaction graph at time t
$v_i \in \mathcal{V}$	Node representing agent i
$(i, j) \in \mathcal{E}_t$	Edge encoding an interaction channel
$\mathcal{N}(i)$	Neighbourhood (set of neighbours) of agent i
$\mathbf{h}^{i,(k)}$	Embedding of agent i after k message-passing rounds
$\psi^{(k)}$	Learnable message function at round k
$\oplus$	Permutation-invariant aggregation operator
$\phi^{(k)}$	Node update function at round k
$\mathbf{e}_{ij}$	Optional edge features for (i, j)
K	Total number of message-passing rounds

5. Benchmarking and Evaluation

This section introduces common benchmarks and evaluation metrics used to assess the performance of multi-agent reinforcement learning algorithms. This has become a rich methodological area of research that has enabled systematic evaluation and hence helped progress in the field. Although some key fundamental developments have been made several years ago, for the sake of completeness we mention key influential works and focus on subfields with most recent developments.

We cover environments, tasks and metrics reported in the literature, from the perspective of four main categories: (1) Evaluation infrastructure, including common evaluation protocols, libraries, reporting experimental pipelines; (2) Core performance evaluation, enabling evaluation of core approaches in a set of shared/common-purpose tasks; (3)

Application-grounded performance, enabling evaluation in specific domains (e.g., embodied robotics, traffic control) among others; and (4) Robustness, generalisation and specialised scenarios, enabling evaluation of approaches in various scenarios beyond the ones observed in training.

Benchmarks provide a standardised approach to compare performance of different algorithms and techniques in specific scenarios. Benchmarks include multi-agent environments and challenges where a variety of competitive, cooperative or mixed-motive settings require agents to learn to address these challenges and achieve environmental goals.

5.1. Evaluation Infrastructure

These works include contributions to the evaluation protocols, infrastructure, libraries and pipelines that enable systematic evaluation of MARL algorithms. They enable interoperability by providing reusable interfaces across different environments. Likewise, facilitating reproducibility by providing standardised evaluation protocols and metrics.

PettingZoo [81] is one of the most influential MARL libraries, providing standard interfaces to multiple environments and tasks, enabling clear, reproducible evaluation of algorithms and extension of existing environments. Likewise, OpenSpiel [82] provides a broad game library together with analysis tools such as NashConv and population-level ranking methods, working not only as an environment collection but also as an evaluation workbench.

Other frameworks improve evaluation by proposing techniques applicable to different problems and domains. For example, a standardised evaluation protocol [83] with concrete defaults for repeated runs, evaluation frequency, and aggregate statistics. Likewise, for scalar returns, authors of [84] propose a meta-game evaluation framework covering multiple environments and making it possible to inspect robustness across self-play, cross-play, social welfare, and strategic dominance relations in the framework. Finally, ref. [85] advanced evaluation through targeted adversarial diagnostics that identify where policies break rather than measure average performance. This approach is especially valuable for applications where catastrophic failures outweigh average-case gains.

EPyMARL [56] emerged to close an early gap in cooperative MARL evaluation: many algorithms claim rested on narrow task sets or incompatible codebases, which made cross-paper comparison unreliable. It addressed that problem by pairing a shared experimental framework with a broad benchmark study across cooperative tasks, giving the field a more consistent empirical baseline. BenchMARL [86] extends the same agenda at a later stage, but shifts the focus from baseline comparison alone to the benchmarking pipeline itself. Its main contribution is to standardise configuration, execution, and reporting across environments, so that cross-task comparison becomes not only possible but easier to reproduce and trust.

JaxMARL [87] closes a related but distinct gap: even when benchmark practice is well structured, evaluation can remain too shallow if training and rollout costs are too high. It addresses this by moving MARL environments and algorithms into an accelerator-native JAX stack, making large multi-seed and multi-task studies much cheaper to run, in turn supporting broader and more statistically reliable evaluation. In that sense, JaxMARL complements rather than replaces EPyMARL and BenchMARL. The earlier two works improve comparability and experimental discipline [56,86] and this one elevates the scale at which that discipline can be applied [87].

A number of metrics are defined within these benchmark solutions, which are used less to define task success than to make empirical comparison trustworthy. The main emphasis is on robust aggregation across seeds, uncertainty quantification, cross-environment sum-

maries, and diagnostics that reveal whether apparent improvements hold under broader statistical or strategic inspection [83–87].

5.2. Core Performance Evaluation

These benchmarks aim to evaluate the effectiveness of MARL algorithms and approaches to converge towards optimal policies that consider key shared multi-agent challenges such as coordination, credit assignment and non-stationarity. There has been a proliferation of benchmarks over the last years, but we cover the most influential ones. Table 3 summarises the evaluation infrastructure metrics used in the literature with respect to MARL.

Table 3. Evaluation infrastructure metrics for multi-agent reinforcement learning.

Metric		What It Measures	Usage Examples
Inter-quartile (IQM)	Mean	Robust aggregate performance across runs, reducing sensitivity to outliers and unusually strong or weak seeds.	Gorsane et al. [83], Bettini et al. [86]
95% confidence intervals/bootstrap intervals	in-	Statistical uncertainty around reported aggregate performance.	Gorsane et al. [83], Bettini et al. [86]
Performance profiles		Fraction of tasks or runs on which a method exceeds a given performance threshold.	Gorsane et al. [83]
Probability of improvement		Likelihood that one method outperforms another across tasks, seeds, or benchmark instances.	Gorsane et al. [83]
Cross-play, self-play, and social-welfare statistics		Strategic relations among policies or algorithms beyond average return alone.	Li and Wellman [84]
Adversarial regret/failure-focused diagnostics		Where a trained policy breaks under targeted hard cases rather than average conditions.	Samvelyan et al. [85]
Wall-clock time/throughput		Practical cost of training and evaluation, especially for large multi-seed or multi-task studies.	Rutherford et al. [87], Bettini et al. [86]

5.2.1. Cooperative Micro-Management

SMAC [88] introduces a multi-agent testbed based on the Real-time strategy game StarCraft II and a set of scenarios that require cooperation among agents to achieve specific objectives. The main underlying model is a decentralised partially observable Markov decision process (Dec-POMDP) where each agent has access to only local observations and must learn to coordinate with other agents to succeed. Although SMAC became popular quickly, it was also found to be possible to solve without achieving robust cooperation, so SMACv2 was introduced, forcing agents to use close-loop, observation-based policies to succeed [89]. Likewise, another extension, SMAC-Hard [90] added opponent diversity and self-play interfaces with the aim of targeting robustness to unseen adversaries.

Cooperation is measured by the win rate on various maps and scenarios. It is possible to evaluate both homogeneous and heterogeneous agent teams, facing different levels of complexity. SMAC family has become a standard benchmark for cooperative multi-agent tasks. Particularly, the benchmark allows for the comparison of diverse methods, ranging from value-decomposition techniques like QMIX [46] and QPLEX [48] to policy-gradient

frameworks like FACMAC [68] and Off-Policy Multi-Agent Decomposed Policy Gradients (DOP) [91], particularly in their ability to handle multi-agent credit assignment [68,91]. Original SMAC is mainly focused on single-objective cooperative settings, MOSMAC [92], a recent extensions have introduced multi-objective MARL scenarios, sequential subtask assignments and temporal horizons that change. Although SMAC enables definition of mixed-motive teams, the benchmark primary enables the evaluation in fully cooperative settings.

5.2.2. Ad Hoc Teamwork and Zero-Shot Coordination

The Hanabi benchmark [93] is based on a cooperative card game of imperfect information for two to five players, where the team's goal is to play cards in five consecutively ordered stacks. The base problem framework is also Dec-POMDP, but its most defining feature is that players can see their teammates' hands but are entirely prohibited from seeing their own cards, creating a setting of intense asymmetric knowledge. Communication is limited to hints about card ranks or colours, which push players to rely on implicit signalling and pragmatic reasoning to succeed. It contributes a unique challenge to the field by mixing communication and environment actions, as every move represent also a communicative intent, instead of having separate communication channel.

The benchmark challenges agents to reason about the hidden beliefs, intentions, and perspectives of their human or artificial partners. The benchmark provides two distinct evaluation tracks: self-play, to find high-utility joint policies, and ad hoc team play, which tests an agent's ability to adapt to unknown partners without pre-coordination. To facilitate this research, the Hanabi Learning Environment (HLE) and high-performance vectorised implementations like JaxMARL offer standardised frameworks for measuring learning efficiency and policy robustness in these complex, non-stationary settings [87,93]. One criticism of Hanabi-like evaluation was that it quickly became saturated with methods achieving near-perfect performance [94]. For this reason, the Yokai Learning Environment [94] introduced tougher challenges such as track beliefs over moving cards, reason under ambiguous hints and decide on game termination based on common ground.

Overcooked [95] is another suite that focuses on ad hoc teamwork and zero-shot coordination. Particularly, assessing performance to partner and layout changes in cooperative settings contextualised in a kitchen environment. These tasks enable robustness evaluation as performance was not tied to specific team and layout configurations. One question that emerged was whether Overcooked, enabled truly evaluation of zero-shot coordination (ZSC), OvercookedV2 [96] address this by adding stochasticity, asymmetric information, communication channels, and scenarios requiring protocol formation at test time, thereby providing a stronger benchmark for ZSC beyond simple state-coverage effects.

5.2.3. Fundamental Coordination in Cross Scenarios

Multi-agent Particle Environments (MPE) [24] enables to define evaluation scenarios where teams-vs-opponents can interact within a particle 2D world. One of the advantages of MPE is its simplicity which enables researchers to focus on agent coordination/anti-coordination and action-level dynamics without the complexity of larger action/state spaces. MPE and derivations have been used to evaluate mechanisms for coordination under non-stationarity [56,97,98], with partial observability [99] and with noisy observations [100].

MPE provides a flexible toolset to define fully cooperative, competitive and mixed-motive scenarios. Given the lightweight nature of the environments and low-dimensional state spaces, it is often used for fundamental testing of algorithms achieving basic behaviour and learning before moving to more complex benchmarks.

5.2.4. Multi-Agent Collective Behaviour

MAgent is a many-agent RL platform [101] that extends core performance evaluation towards large-population settings by providing a platform for studying many-agent interaction at scale. Rather than focusing on small teams or narrowly defined co-operative tasks, it enables evaluation of emergent collective behaviour in environments where large numbers of agents must compete, co-operate, or form population-level strategies under local interaction rules. This makes it a useful benchmark for assessing how MARL methods scale beyond few-agent settings and whether they can capture artificial collective intelligence rather than only pairwise or team-level co-ordination.

Core performance benchmarks mainly measure whether agents solve the task, how reliably they do so, and how efficiently learning progresses. Although the exact task score differs across suites such as SMAC, Hanabi, Overcooked, Melting Pot, and MPE, the dominant pattern is to combine a task-success metric with episodic reward and a notion of learning speed or sample efficiency [56,88,93,102]. The most common metrics used, a brief description and examples of where these are used is presented in Table 4.

Table 4. Core performance metrics for multi-agent reinforcement learning.

Metric		What It Measures	Usage Examples
Win rate/success rate		The fraction of evaluation episodes in which the team completes the scenario or defeats the opponent.	Samvelyan et al. [88], Ellis et al. [89], Kurach et al. [103]
Episode turn/cumulative reward	re-	Total team reward over an episode, often averaged across tasks or seeds.	Papoudakis et al. [56], Bettini et al. [86]
Task score		Domain-specific task performance within a shared benchmark, such as bounded game score or goal difference.	Bard et al. [93], Gessler et al. [96], Kurach et al. [103]
Perfect-game rate/solved-task rate		Fraction of runs or episodes that reach the maximum or fully solved outcome.	Bard et al. [93], Ruhdorfer et al. [94]
Learning speed/sample efficiency	effi-	How quickly stable or target performance is reached as environment interaction increases.	Papoudakis et al. [56], Samvelyan et al. [88], Rutherford et al. [87]
Scenario-wise or map-wise aggregation	map-	Whether performance holds consistently across multiple scenarios rather than one headline task.	Samvelyan et al. [88], Leibo et al. [102]

5.3. Robustness, Generalisation and Specialised Evaluation

These benchmarks are designed to evaluate the robustness of MARL algorithms and approaches to changes in the environment, agent population, and task conditions, as well as their ability to adapt to new scenarios and partners.

5.3.1. Offline MARL

Offline multi-agent reinforcement learning studies how multiple agents can learn coordinated policies from a fixed dataset of previously collected interactions, without further access to the environment during training [104]. It is motivated by settings where online exploration is costly, unsafe, or impractical, so progress depends on learning effectively from static logged data rather than continual trial-and-error interaction. Off-the-Grid MARL [105] addresses the lack of shared datasets, baselines, and evaluation protocols

for offline multi-agent reinforcement learning by providing a framework for generating offline datasets from cooperative MARL environments together with standard baselines and evaluation procedures. A key aspect of this setting is that the benchmark is shaped not only by the task itself, but also by the quality, diversity, and bias of the logged data, making dataset generation a central part of evaluation.

Building on this, ref. [106] shows that much reported progress in offline MARL is difficult to trust because baselines, training budgets, environment choices, and reporting practices have often been inconsistent. By demonstrating that simple, well-implemented baselines can match or exceed purported state of the art on many published datasets, this work reframes offline MARL benchmarking as a question of methodological rigour as much as algorithm design, which is particularly important in application areas where learning from static logs is more realistic than online interaction.

5.3.2. Zero-Shot Coordination

Apart from the evolution core performance evaluation suites such as OvercookedV2 [96] and Yokai [94] that were designed to address zero-shot coordination over existing influential benchmark suits. ZSC-Eval [107] addresses a key limitation in zero-shot coordination benchmarking by showing that evaluation depends not only on the environment, but also on how partner sets are constructed. Rather than testing agents against a small or arbitrarily chosen set of partners, it defines a fuller evaluation pipeline in which candidate partners are generated, a diverse subset is selected through best-response diversity, and generalisation quality is measured through best-response proximity. This makes partner diversity and partner difficulty explicit components of the benchmark, which is particularly relevant for human-AI coordination, ad hoc teamwork, and other settings where agents must cooperate reliably with previously unseen partners.

5.3.3. Social Generalisation

Melting Pot [102] is a rich evaluation suite for MARL built to assess social generalisation across a wide range of scenarios mixing cooperative, competitive, and mixed-motive interactions. Unlike benchmarks centred on isolated team tasks, Melting Pot focuses on whether agents can perform effectively in novel social situations, especially when interacting with previously unseen partners or populations.

Its core contribution is to evaluate robustness to unfamiliar conventions, partner behaviours, and incentive structures rather than performance in self-play alone. Scenarios are formalised through the combination of a substrate and a reference background population, allowing systematic evaluation of how well policies transfer to new social partners and interaction settings [102]. In this sense, Melting Pot is best understood as a benchmark for social robustness and generalisation, rather than only for core task performance.

Melting Pot 2.0 [108] expands this framework by increasing the number of substrates and scenarios, adding asymmetric roles, and further standardising evaluation around reusable reference settings. This strengthens its role as a benchmark for generalisation to novel social partners and mixed-incentive interaction settings.

5.3.4. Transfer Across Environment

Built on top of a grid-world setting for social learning [109], HECOGGrid [110] proposes a family of MARL environments in which two key factors can be varied systematically: coordination requirements and environmental heterogeneity. Coordination requirements are controlled by the number of agents needed to complete a task, such as collecting a treasure, while heterogeneity is controlled by the extent to which properties of the environment vary across regions, including spatial layout, obstacles, and transition dynamics.

Because HECOGrid provides a parameterised family of environments rather than a fixed set of maps, it supports evaluation across both training and held-out variants. This makes it particularly useful for studying transfer and out-of-distribution generalisation across different coordination and heterogeneity levels, and therefore places it most naturally under robustness, generalisation, and specialised evaluation settings [110].

5.3.5. Open-Ended Horizon Generalisation

Craftax-MA [111] extends the single-agent open-ended RL environment Craftax [112] to the multi-agent setting, enabling long-horizon interaction in procedurally generated worlds with resource gathering and survival-style mechanics. It is motivated by the limitation of many MARL benchmarks towards narrow, short-horizon tasks that place relatively little pressure on long-term dependencies, exploration, and generalisation.

Instead, Craftax-MA provides a single benchmark family spanning a range of general abilities, including exploration, resource competition, and coordination. A more demanding variant, Craftax-Coop, further introduces heterogeneous roles such as Forager, Miner, and Warrior, together with trading and cooperation-critical mechanics. Because evaluation takes place over a broad distribution of procedurally generated worlds, agents must perform robustly across varying layouts, resource placements, and long-horizon contingencies, making Craftax-MA best suited to the category of robustness, generalisation, and specialised evaluation settings [111]. Specific metrics for robustness and adaptability are introduced to measure adaptability of agents to various conditions, a selection of these metrics is presented in Table 5.

5.3.6. Continual MARL

MEAL [113] addresses the lack of a shared benchmark for continual multi-agent reinforcement learning, a setting that had remained underexplored despite the existence of both continual learning and MARL benchmarks in isolation. It introduces the first benchmark designed specifically for this intersection and uses an accelerator-friendly implementation to support long task sequences, making continual evaluation practical at a scale where forgetting and adaptation failures become visible. This is particularly relevant for multi-agent systems that must operate across changing deployments over time, such as service robotics, adaptive game agents, and operational control systems that need to adapt across layouts, seasons, or missions without retraining from scratch.

5.3.7. Multi-Objective MARL

MOMAland [114] addresses the fact that many realistic multi-agent systems cannot be evaluated through a single scalar objective, as domains such as traffic systems, power grids, and infrastructure management often require balancing efficiency, fairness, safety, cost, and environmental goals. To support this setting, it standardises multi-objective multi-agent reinforcement learning through a library of environments, baseline algorithms, and compatibility wrappers, turning the intersection of MARL and multi-objective RL into a coherent benchmark regime rather than a collection of isolated case studies. This is especially important in applications involving multiple stakeholders or normative trade-offs, where collapsing all objectives into a single reward too early can obscure the real evaluation problem.

5.3.8. Competitive MARL and Exploitability

FightLadder [115] addresses a longstanding gap in competitive multi-agent reinforcement learning by providing a lightweight open benchmark with rich visual dynamics and evaluation beyond raw win rate. It introduces a real-time fighting-game platform together with Elo-style ranking and exploitability tests, making it possible to assess not only whether

an agent performs well on average, but also whether it can be systematically exploited by a dedicated opponent. This is particularly relevant in strategic settings involving adaptive adversaries, where average performance alone may obscure brittle policies that are unsuitable for security, competitive games, or adversarial planning.

Table 5. Robustness, generalisation, and specialised-scenario metrics for multi-agent reinforcement learning.

Metric	What It Measures	Usage Examples
Cross-play performance/SP–XP gap	How much performance degrades when paired with unseen partners rather than self-play partners.	Gessler et al. [96], Ruhdorfer et al. [94]
BR-Div and BR-Prox	Diversity of evaluation partners and proximity to best-response behaviour under zero-shot co-ordination.	Wang et al. [107]
Held-out environment or heterogeneity performance	Transfer to unseen layouts, maps, or heterogeneity levels.	Liu et al. [110], Leibo et al. [102], Omari et al. [111]
Offline normalised return/dataset-wise aggregate performance	Performance from static datasets under fixed training and evaluation protocols.	Formanek et al. [105], Formanek et al. [106]
Forgetting/forward transfer/backward transfer	Retention and adaptation across long task sequences.	Tomilin et al. [113]
Hypervolume/Pareto coverage/objective trade-off quality	How well a method balances multiple objectives rather than one scalar reward.	Felten et al. [114]
Exploitability/Elo/adversarial robustness	Whether a policy remains strong against adaptive opponents rather than only on average.	Li et al. [115], Samvelyan et al. [85]

5.3.9. Robustness and Social Dilemmas

MADRID [85] takes a failure-oriented view of MARL benchmarking by generating diverse adversarial scenarios designed to expose weaknesses in trained policies, rather than relying only on average-case performance over a fixed suite. Its main contribution is to make robustness diagnosis itself a benchmarkable object, which is particularly relevant in mixed cooperative–competitive settings where strong average performance can still conceal large behavioural blind spots.

SocialJax [116] complements this perspective by showing that specialised social-dilemma evaluation also depends on efficient systems support. By reimplementing sequential social-dilemma benchmarks in JAX, it enables broader and more statistically reliable large-scale studies while retaining task-specific metrics tied to prosocial and antisocial behaviour. This makes it useful for applications such as social simulation, collective action, and the study of mechanisms for promoting cooperation in large populations.

Metrics in this category are more fragmented because they follow the failure mode under study. Zero-shot co-ordination metrics focus on unseen partners, social-generalisation suites focus on transfer across populations or situations, offline MARL cares about reliability across datasets and baselines, continual benchmarks track forgetting and adaptation, and competitive or multi-objective settings need dedicated measures such as exploitability or Pareto-aware aggregates [106,107,113–115]. A selection of these metrics is presented in Table 5.

5.4. Application-Grounded Benchmarks

These benchmarks provide environments, scenarios, test suites, and metrics to evaluate MARL performance in specific domains. Due to the diversity of domains and specific suites in each one, we briefly cover a selection of the most recent benchmarks in key application domains. Main metrics for each application domain are presented in Table 6.

5.4.1. Multi-Robot Coordination, Embodied Collaboration, and Sim-to-Real Evaluation

Embodied robotics is the second major centre of benchmark development, but recent work has diversified its scope considerably. Multi-Agent MuJoCo remains an influential reference point because it established decentralised continuous co-operative control in physics-based multi-agent settings [117]. VMAS follows a more scalable design, using a vectorised 2D physics engine and a set of collective robot-learning scenarios to support efficient comparative evaluation across algorithms [118]. Both are useful because they capture central control and co-ordination demands without requiring the full cost of high-fidelity robotics simulation.

More recent benchmarks push further towards richer embodied interaction. SMART combines an integrated simulator with a real multi-robot system and demonstrates the setup on co-operative driving-style scenarios, thereby positioning the environment as both a training and evaluation platform [119]. NeuronsMAE similarly provides configurable co-operative and competitive multi-robot tasks and explicitly uses parameter variation to support transfer and sim-to-real analysis [120]. IsaacLab-based work takes the next step towards high-fidelity robotics benchmarking: IsaacLab-HARL extends Isaac Lab to heterogeneous multi-agent adversarial reinforcement learning, while related work on heterogeneous collaboration in Isaac Lab focuses on scalable simulation for robotic co-operation rather than simplified benchmark tasks alone [121–123].

A related stream concentrates on more structured embodied collaboration. RoboFactory introduces compositional manipulation constraints that make embodied multi-agent collaboration more systematically benchmarkable [124]. PARTNR shifts attention towards planning and reasoning in embodied multi-agent tasks, while VIKI-R studies embodied co-operation more directly through reinforcement learning [125,126]. MQE contributes a quadruped-centred benchmark where interaction is itself the key difficulty, and Assistax specialises further into assistive robotics through a hardware-accelerated benchmark stack [127,128]. AT-Drone also fits naturally within this category, as its main emphasis is adaptive teaming among embodied aerial agents rather than pure path planning [129].

Another important development is the growing focus on sim-to-real evaluation. MARBLER bridges the Robotarium physical testbed with a Gym-style interface, enabling reproducible deployment and benchmarking on real robot collectives within a familiar MARL workflow [130]. JaxRobotarium targets the same niche but emphasises speed and scalability through JAX acceleration, making rapid training and deployment more practical while retaining the constraints needed for realistic transfer [131]. Shared human spaces introduce a further layer of difficulty. SOCIALGYM 2.0 and Arena 4.0 both target multi-robot learning in human-centred navigation settings, with Arena moving more explicitly towards ROS2-based and deployment-oriented evaluation [132,133].

5.4.2. Path Planning, Routing, and Warehouse Logistics

These benchmark suites focus on instance generation and standardised protocols for multi-agent navigation problems at scale. More broadly, path planning and routing remain a coherent benchmark family because they support controlled instance generation, strong domain metrics, and direct comparison between learned and planning-based methods. POGEMA [134] is a platform for cooperative multi-agent pathfinding; it includes a large-

scale grid simulator, problem-instance generator, predefined instance sets, visualisation, and an automated benchmarking tool with domain metrics, supporting generalisation to unseen maps. CAMAR [135] extends this line of work to continuous-space, continuous-action multi-agent routing, aiming to provide a reproducible benchmark suite with scenario sets, high simulation throughput, and a structured evaluation protocol. GRACE [136] broadens the same agenda by unifying grid, roadmap, and continuous path-planning settings within a single simulator and benchmarking framework.

More specialised benchmark solutions target logistics and industrial settings. URB introduces a dedicated benchmark for connected autonomous vehicle routing, while MRTA-Sim brings together task allocation, planning, and control in open-world multi-robot settings [137,138]. Warehouse-oriented benchmarks and datasets contribute more structured fulfilment settings, including large-scale MAPF instances and realistic industrial path-planning tasks [139,140]. A lighter continuation of this line adapts one-shot multi-agent pathfinding from gridworlds to AGV warehouse settings, keeping evaluation tractable while moving closer to deployment constraints [141]. A more specific benchmark solution for warehouse environments is presented in RWARE (Multi-Robot Warehouse), included as part of the EPyMARL suite [56]. This open-source environment stresses coordination in warehouse-style navigation and manipulation workflows, enabling systematic and reproducible comparisons across multiple cooperative tasks and algorithms under a consistent evaluation pipeline.

5.4.3. Autonomous Driving, Traffic Control, and Transport Systems

Aiming to provide evaluation suites closer to real-world traffic and transport control, recent benchmark work has expanded from stylised driving tasks towards data-driven and closed-loop settings. General urban driving benchmarks broaden the range of tested traffic situations, while TorchDriveEnv adds reactive and diverse non-playable characters to make interaction a more central part of learning and evaluation [142,143]. In a similar direction, NAVSIM provides a data-driven simulation stack for autonomous-vehicle benchmarking, whereas Bench2Drive turns closed-loop end-to-end driving into a multi-ability benchmark, making it possible to evaluate perception, planning, and control jointly rather than through a single route-level score [144,145].

Recent work has also pushed strongly on scalability. GPUDrive enables high-throughput, data-driven multi-agent driving simulation, making broad training and evaluation over many scenarios practical at a scale that earlier road simulators could not support [146]. MDrive focuses more specifically on closed-loop co-operative driving for end-to-end multi-agent systems, addressing the need for benchmarks in which interaction among learning agents is itself a first-class challenge [147]. Within this broader transport context, IntersectionZoo [148] introduces a cooperative eco-driving application that combines partial observability, multi-objective trade-offs, and heterogeneous real-world conditions. The suite provides a micro-traffic simulation environment with around one million traffic scenarios involving factors such as temperature, traffic conditions, and travel demand, thereby shifting evaluation towards generalisation across heterogeneous operating conditions.

Traffic control forms a related but distinct line of work because the target is network-level co-ordination rather than vehicle-level decision-making. PyTSC provides a unified platform for MARL in traffic signal control, while VissimRL places the same problem in a widely used microscopic traffic simulator and therefore moves closer to evaluation settings with operational relevance [149,150]. Taken together, these works show that transport benchmarks now span several levels of control, from vehicle interaction and end-to-end driving to decentralised signal management.

5.4.4. Energy, Power, and Infrastructure Systems

Energy and infrastructure benchmarks are less numerous than those in transport or robotics, but they are becoming more mature and methodologically structured. Common-Power illustrates this clearly by providing a framework for safe, data-driven smart-grid control that supports comparison across learning-based and control-based approaches rather than MARL alone [151]. RL2Grid goes further by defining standard tasks and safety-aware evaluation for power-grid operations, making it easier to compare approaches in a domain where operational assumptions often differ substantially across studies [152].

Recent work also broadens the range of infrastructure settings covered by MARL benchmarks. EnEnv 1.0 is designed explicitly as an energy-grid environment for MARL evaluation, while MARLEM shifts the focus to decentralised local energy markets, where implicit co-operation among agents becomes a key part of the benchmark itself [153,154]. MuFlex studies coordinated flexibility across multiple buildings, and OpenG2G extends the same general logic to datacentre-grid runtime co-ordination [155,156]. In parallel, work on multi-objective reinforcement learning with shielding in power grids highlights an important feature of this category: realistic evaluation often depends on treating safety and competing objectives as part of the benchmark, rather than as constraints added after optimisation [157].

5.4.5. Sports and Team-Play Environments

Sports environments form a smaller but still relevant benchmark family because they combine continuous control, co-ordination, and long-horizon team structure within a single domain. Google Research Football remains the main reference point, providing both full-game settings and smaller football-academy tasks that support evaluation at different levels of complexity [103]. RobocupGym continues this line in a robot-soccer setting that stresses continuous control more explicitly, while MARLadona moves towards a more co-operative team-play framing in a football-like environment [158,159].

Although this category has seen less recent diversification than driving or robotics, it remains useful as a middle ground between highly abstract co-ordination tasks and heavily engineered real-world simulators. In practice, these environments provide a convenient domain for evaluating timing, positioning, and partner adaptation before moving to settings with stronger safety or deployment requirements [103,158,159].

5.4.6. Others

A small number of recent application-grounded benchmark proposals do not fit naturally into the other categories. Multi-Agent Craftax is best understood as an open-ended benchmark for long-horizon, procedurally generated multi-agent interaction rather than a benchmark tied to a single application domain [111]. FightLadder similarly contributes a strong competitive MARL benchmark built around fighting games and exploitability-aware evaluation, but it is better viewed as part of competitive-game benchmarking than as one of the main application-grounded families covered above [115].

A further emerging niche is space and extra-terrestrial robotics, where benchmark design is beginning to reflect the distinctive constraints of orbital dynamics, microgravity operations, and off-world manipulation. Space Robotics Bench proposes a robotics benchmark suite explicitly aimed at robot learning beyond Earth, making it one of the clearest efforts to turn space robotics into a reusable evaluation setting rather than a collection of bespoke tasks [160]. OrbitZoo plays a related role for orbital control by framing real orbital-system challenges as reinforcement learning benchmarks, while SmallSatSim focuses more narrowly on high-fidelity simulation and training for microgravity robotic close-proximity operations [161,162]. Together these works suggest that space robotics is

becoming a recognisable application niche for MARL evaluation, although it remains less mature and less standardised than the main benchmark families discussed above.

Table 6. Application-grounded performance metrics for multi-agent reinforcement learning.

Metric	What It Measures	Usage Examples
Route completion/task completion	Whether the agent or team reaches the goal under realistic environment constraints.	Jia et al. [145], Qin et al. [124]
Collision rate/safety violations	Frequency of unsafe interactions with vehicles, robots, humans, or obstacles.	Jia et al. [145], Lavington et al. [143], Sprague et al. [132], Eichelbeck et al. [151]
Safety margin/near-miss rate	How often agents operate close to unsafe states even when no direct collision occurs.	Jia et al. [145], Demirel et al. [157]
Energy consumption/resource usage	Amount of energy, fuel, or other consumable resources required to complete the task.	Jayawardana et al. [148], Marchesini et al. [152], Wu et al. [155]
Latency/response time	Delay between observation, decision, and effective action, or time required to complete critical control responses.	Chang et al. [150], Chung et al. [156]
Communication load/bandwidth usage	Amount of information exchanged among agents or between agents and infrastructure during task execution.	Qin et al. [124], Kang et al. [126]
Travel time/delay/queue length	Efficiency of flow through traffic networks or routing systems.	Bokade and Jin [149], Chang et al. [150]
Path length/makespan/coordination efficiency	Quality and efficiency of joint navigation or routing solutions, often alongside success rate.	Skrynnik et al. [134], Pshenitsyn et al. [135], Wang et al. [139]
Fairness/service balance	Whether benefits, delays, or resources are distributed equitably across agents, users, or controlled entities.	Felten et al. [114], Salazar-Peña et al. [154]
Robustness under disturbances	Stability of performance under noise, failures, incidents, or changing operating conditions.	Jayawardana et al. [148], Chang et al. [150], Beukman et al. [158]
Constraint satisfaction	Degree to which task, safety, physical, or operational constraints are respected during execution.	Qin et al. [124], Marchesini et al. [152], Demirel et al. [157]
Energy cost/peak load/ramping/load factor	Operational efficiency and stability in energy systems.	Marchesini et al. [152], Eichelbeck et al. [151]
Goal difference/match outcome	Team-play effectiveness in sports environments with competitive structure.	Kurach et al. [103], Beukman et al. [158], Li et al. [159]
Domain-specific key performance indicators	Task-specific operational indicators that are meaningful only within the target application domain.	Eichelbeck et al. [151], Jia et al. [145], Jayawardana et al. [148]

6. Recent Advances and Open Challenges

This section overviews the advancements and the open problems in the field categorised in different respects, which have been provided in the following subsections. The section begins with a brief discussion of the development of MARL research directions, benchmarks, and evaluation practices over time. It provides a high-level overview of the major advancements that have shaped the field, together with the key challenges and open problems that continue to motivate ongoing research.

6.1. A Historical Overview of MARL Research Themes and Evaluation

Figure 4 provides an evolutionary overview of the MARL landscape and situates the major themes covered in this survey within their historical context. The figure illustrates the progression from early independent learning approaches toward centralised training with decentralised execution (CTDE), value decomposition, actor–critic methods, communication learning, graph-based architectures, and more recent model-based and meta-learning approaches. It also highlights the parallel evolution of benchmark environments and

evaluation methodologies, reflecting the growing emphasis on robustness, generalisation, reproducibility, and real-world deployment. This progression motivates the structure of the survey and provides context for understanding how current MARL research has emerged from earlier developments.

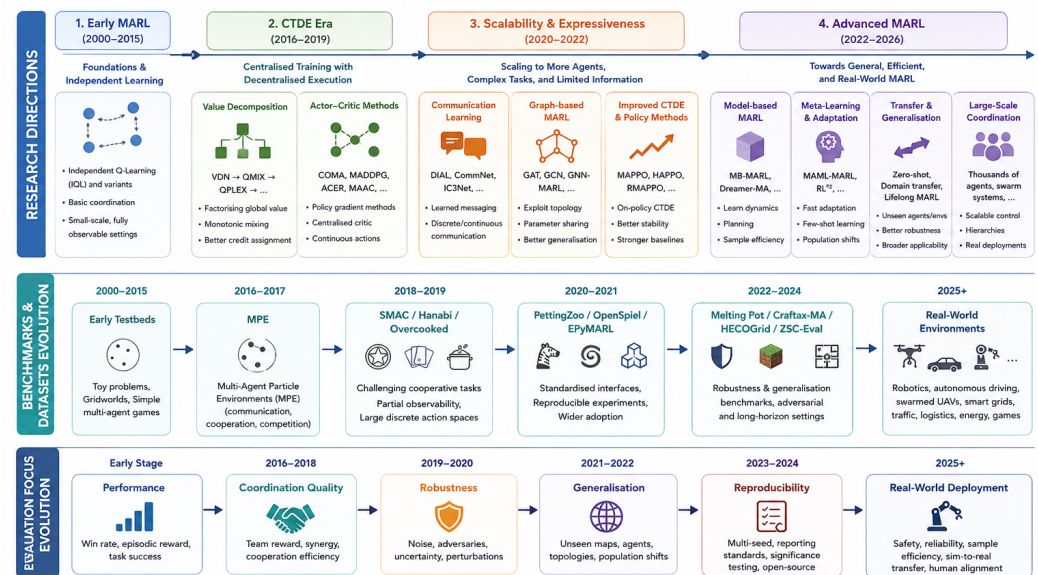


Figure 4. Evolution of MARL research methods, benchmarks, and evaluation practice over time. The figure summarises the major stages in the development of MARL.

6.2. Advances in Algorithms and Architectures

Algorithmic progress across the families surveyed in Section 4 has proceeded along several interconnected axes. Within value decomposition, QPLEX [48] demonstrated that full IGM-compliant factorisation is achievable without QMIX’s monotonicity restriction via a duplex duelling architecture, whilst Sun et al. [10] extended distributional RL [40] to cooperative factorisation, showing that modelling return uncertainty at the factorisation level improves credit assignment in stochastic cooperative tasks. MAVEN [163] introduced a shared latent variable to encourage diverse strategy discovery, directly addressing the tendency of value decomposition methods to converge to suboptimal equilibria. On the policy gradient side, HAPPO [69] provided the first formal monotonic improvement guarantee for heterogeneous cooperative MARL through a sequential trust-region update scheme, and FACMAC [68] demonstrated that applying policy gradients through a factorised centralised critic yields substantial coordination gains at higher agent counts compared with independent per-agent critics. Attention-based architectures such as MAAC [67] and Qatten [164] leverage self-attention to selectively aggregate information across agents, improving scalability in larger populations. On the communication front, TarMAC [71] learns when and what to communicate under bandwidth constraints, extending CTDE to partially decentralised execution. Critically, systematic benchmarking [56] and the implementation analysis of Engstrom et al. [55] jointly confirm that performance rankings across algorithmic families are more sensitive to implementation choices and task structure than to fundamental algorithmic distinctions—a finding that demands controlled ablation studies rather than headline comparisons. Graph-based representations have emerged as a principled architectural response to the scalability limitations of flat joint-input methods, with DGN [76], DCG [78], and DICG [80] collectively establishing that GNN-based value estimators outperform flat-input counterparts as agent populations grow, and an emerging frontier in the application of diffusion-based generative models to joint action sampling,

providing a principled mechanism for representing complex multi-modal coordination distributions [98].

6.3. Scalability to Large-Scale Systems

The CTDE paradigm addresses non-stationarity by conditioning the critic on the global state and joint actions during training, which mitigates some of its effects by stabilising the learning signal rather than providing the environment fully stationary from each agent's optimisation perspective [24]. Crucially, this computational overhead is confined entirely to the offline training phase: during deployment, each agent executes its policy independently using only local observations; for simple independent actors this yields approximately constant per-agent complexity and supports real-time decision-making in large-scale systems, although in general the per-agent cost depends on the observation size, communication neighbourhood, recurrent history length, graph message-passing depth, and attention aggregation used by the executable policy [16]. This contrasts with centralised execution, wherein a central controller creates communication bottlenecks and single points of failure. Independent learning retains clear complementary scalability advantages in settings involving hundreds or thousands of agents, competitive environments where sharing training information is infeasible, and deployment contexts where communication infrastructure is unavailable; its minimal implementation overhead and constant per-agent training complexity continue to make it the default choice for computationally constrained applications [34,56]. As agent populations grow, however, centralised critics conditioning on the full joint action space face exponential input dimensionality growth. Value decomposition (VDN, QMIX) and attention aggregation (MAAC) are the primary architectural mitigations, though the representational constraints of additivity and monotonicity may limit expressiveness in complex coordination tasks [48,163]. Graph-based methods offer a structured approach to scalability: the locality inductive bias of message passing restricts each agent's computation to a K -hop neighbourhood regardless of total system size, and permutation equivariance enables zero-shot generalisation to populations of different sizes and orderings—a property unattainable by fixed-size joint-input architectures [76,78]. These benefits are nonetheless contingent: performance is sensitive to the choice of interaction graph, deep message passing can induce over-smoothing of agent embeddings, attention-based aggregation adds computational cost, and inferring a useful graph under partial observability remains difficult. Model-based approaches face a compounding challenge: as agent policies co-evolve, joint dynamics change continuously, degrading model accuracy in a self-reinforcing feedback loop; these methods remain most reliably beneficial when agent interactions are sparse and cooperative structures evolve slowly [57]. Despite methodological progress across all families, most reported benchmarks involve fewer than 30 agents in simulated environments, and convincing demonstrations at the scale of hundreds of agents under real-time constraints remain limited [21].

6.4. Real-World Deployment and Safety Concerns

The CTDE paradigm aligns naturally with a wide range of real-world problem structures, exploiting the asymmetry between information-rich simulation at training time and partial observability under deployment. The level of real-world validation nevertheless differs markedly across the application domains surveyed below, ranging from genuinely operational systems to results established only in simulation; we therefore indicate the evidence level in each case. In multi-robot coordination, CTDE methods have enabled emergent tool use and sophisticated collaborative strategies without explicit reward shaping, albeit within simulated physics environments [165]. In autonomous vehicles, centralised training in high-fidelity simulators has yielded policies that, in simulation, coordinate lane

changes, merging, and intersection traversal using only onboard sensors; this evidence derives from microscopic traffic simulation frameworks (e.g., the SUMO-based Flow) rather than on-road deployment [166,167]. The StarCraft Multi-Agent Challenge (SMAC) [88] has served as a canonical benchmark, with algorithms such as QMIX and QTRAN achieving high win rates on several scenarios; SMAC is, however, a simulated benchmark rather than a deployed system. Application to network optimisation and resource allocation—including traffic signal control [168], wireless spectrum management [169], and data centre cooling [170]—further demonstrates the paradigm’s versatility, though the evidence level varies by domain: traffic signal control and wireless power allocation are evaluated in simulation (using the CityFlow traffic simulator and a simulated cellular network, respectively), whereas the data-centre cooling study reports a genuinely operational result—obtained, however, with a single centralised model-predictive controller rather than a fully decentralised multi-agent policy. CTDE has also been explored for human-AI collaboration in search and rescue, logistics, and healthcare, where agents must anticipate and adapt to human behaviour using centralised training information [171,172]; this work has so far been confined to simulation and controlled laboratory studies rather than operational deployment. Model-based methods are most advantageously deployed in sample-scarce contexts—physical robot learning, safety-critical systems, and settings with expensive simulation—where the overhead of model learning is offset by reductions in real-world interaction cost [57]. Across all families, a persistent deployment concern is generalisation under distributional shift: policies trained in simulation may perform poorly when environment dynamics at deployment differ from training conditions. Domain randomisation [173], adversarial training [174], and meta-learning [175] offer partial remedies without resolving the challenge comprehensively.

Robustness of coordination has gained increasing attention recently. Coordination is still brittle to configuration changes in agent teams and environment dynamics. In Dec-POMDP settings with non-adaptive partners, agents learn based on co-adapted agreements rather than on observations, leading to policies that fail when partners change [176]. Researchers have identified vulnerabilities of MARL approaches to direct adversarial attacks, demonstrating that current approaches remain prone to fail under systematic perturbations [177]. Likewise, evaluation platforms are integrating multiple attack surfaces at policy, state, action, reward, or environment levels to facilitate robustness evaluation and adversarial vulnerability assessment [178].

6.5. Open Problems and Future Research Directions

Although MARL and CTDE methods have demonstrated strong empirical performance across benchmark environments, major open problems remain across theory, scalability, robustness, communication, generalisation, reproducibility, and deployment. These challenges differ substantially in maturity and evidence requirements, suggesting the need for a more structured research agenda.

6.5.1. Theory and Learning Guarantees

Theoretical understanding remains substantially behind empirical progress. CTDE improves training stability by mitigating decentralised non-stationarity, and value decomposition methods provide consistency guarantees under specific assumptions such as monotonicity [46,47]. Independent learning approaches have convergence results only in restricted settings, including matrix games for hysteretic Q-learning [34]. Recent analyses also identify policy-gradient non-stationarity arising from simultaneous policy updates as a distinct challenge in multi-agent optimisation [16,69]. General convergence guarantees for MARL in stochastic sequential environments remain unavailable. It is still unclear

under what conditions independent learning is sufficient, how actor–critic methods behave under decentralised execution constraints, and whether graph-based representations satisfy formal properties such as individual-global-max (IGM) consistency [44,48,78]. Sample complexity and stability bounds for modern MARL algorithms also remain poorly characterised [12,21]. Progress requires formal analyses establishing convergence conditions, finite-sample guarantees, and theoretical comparisons between value-factorisation, policy-gradient, and graph-based approaches under controlled assumptions.

6.5.2. Scalability and Coordination

MARL systems increasingly scale through value factorisation, parameter sharing, and graph-based coordination mechanisms [45,76]. Approaches such as QMIX achieve effective coordination in cooperative environments with monotonic reward structures [179]. Scalable credit assignment under sparse, delayed, or non-monotonic rewards remains difficult [48]. Coordination quality often degrades as agent populations increase, and graph topology selection is still largely manual and environment-dependent [7]. Existing methods also provide limited understanding of how coordination complexity grows with team size. Future work should evaluate algorithms across varying numbers of agents, communication constraints, and reward structures, supported by scaling laws and computational efficiency analyses.

6.5.3. Robustness and Safety

Centralised critics and coordinated training can improve robustness to local uncertainty and partial observability [24]. Empirical studies show increased stability for some policy-gradient methods such as MAPPO in settings where monotonic factorisation becomes restrictive [54,180]. Current methods remain sensitive to environment shifts, adversarial interactions, partial observability, and failures of individual agents [176–178]. Safety guarantees under decentralised execution and mixed cooperative–competitive environments remain limited [181]. Research should prioritise robustness benchmarks involving distribution shifts, adversarial perturbations, communication failures, and explicit safety metrics rather than average episodic return alone.

6.5.4. Communication and Representation Learning

Learned communication and graph-based interaction models improve coordination efficiency and reduce reliance on handcrafted communication protocols [23,71,76]. The mechanisms by which communication structures emerge and adapt remain insufficiently understood [7]. Automatic graph construction, communication sparsity, and the representation of dynamic cooperative and adversarial relationships remain open questions [79,80]. Progress requires controlled ablation studies, interpretable communication analyses, and evaluation across heterogeneous communication environments.

6.5.5. Generalisation and Transfer

Most current MARL methods achieve strong performance within fixed training distributions and benchmark environments [56,89]. Cross-task transfer, adaptation to changing agent populations, and generalisation beyond benchmark-specific dynamics remain limited [102,175,182]. Many reported gains do not consistently transfer across domains [56]. Future evaluation should include out-of-distribution testing, transfer benchmarks, curriculum-based experiments, and long-horizon adaptation studies.

6.5.6. Reproducibility and Evaluation Methodology

Benchmark suites for cooperative MARL continue to mature and have improved comparability across studies [56,86]. Meta-analyses identify persistent weaknesses in-

cluding inconsistent baselines, insufficient reporting standards, limited statistical rigour, and inadequate attention to factors such as over-coordination and complementary observability [183,184]. Future work should adopt standardised evaluation protocols, multiple random seeds, significance testing, open-source implementations, and benchmark governance practices [83,185].

6.5.7. Real-World Deployment

CTDE assumes access to privileged state information during training, an assumption that often simplifies controlled experimentation [24]. Many practical settings lack global state access, exhibit partially unknown dynamics, or require online adaptation [7,17,186]. The gap between benchmark success and deployment reliability remains substantial [13]. Advances require real-world evaluations, resource-constrained experiments, deployment case studies, and methods that operate under limited observability and incomplete training information.

Overall, the field's central challenge is no longer demonstrating isolated empirical improvements, but establishing whether MARL methods can scale, generalise, remain robust, and be reproduced reliably outside benchmark environments.

7. Conclusions

This paper has presented a narrative survey of deep reinforcement learning approaches for multi-agent systems, with particular focus on recent advances in multi-agent deep reinforcement learning. The survey makes four principal contributions. First, it formalises the central challenges of MARL—including non-stationarity, partial observability, credit assignment, and scalability—within a unified problem setting grounded in the Dec-POMDP framework. Second, it classifies and critically evaluates recent algorithmic advances spanning centralised training with decentralised execution, value decomposition, policy gradient and actor–critic frameworks, model-based approaches, graph-based architectures, and learned communication. Third, it reviews empirical benchmarks and evaluation practices across systemic performance, robustness and adaptability, and domain-specific evaluation. Fourth, it identifies key open problems and promising future directions, including theory and learning guarantees, scalability and coordination, robustness and safety, communication and representation learning, generalisation and transfer, reproducibility and evaluation methodology, and real-world deployment.

Across the algorithmic families surveyed, the centralised training with decentralised execution (CTDE) paradigm has emerged as the dominant framework for cooperative MARL, underpinning value decomposition methods, multi-agent actor–critic architectures, and graph-based value estimators alike. Within this paradigm, approaches such as QMIX and QPLEX have advanced the expressiveness of cooperative value factorisation, whilst MAPPO and HAPPO have established that on-policy policy gradient methods are more competitive in cooperative settings than previously assumed. Graph-based architectures have emerged as the most principled response to the scalability limitations of flat joint-input representations, exploiting relational inductive biases that enable generalisation across varying agent populations. Independent learning, despite its theoretical limitations, retains practical relevance in large-scale and communication-constrained deployments where centralised training information is unavailable.

The survey reviewed benchmarking and evaluation practices across four complementary categories: evaluation infrastructure, core performance benchmarks, robustness and generalisation benchmarks, and application-grounded evaluation. Evaluation infrastructure frameworks such as PettingZoo, OpenSpiel, EPyMARL, BenchMARL, and JaxMARL have played a critical role in improving reproducibility, comparability, and statistical rigour

by providing standardised interfaces, protocols, and reporting methodologies. Core performance benchmarks, including SMAC, Hanabi, Overcooked, and MPE, have become canonical testbeds for assessing fundamental MARL challenges such as coordination, credit assignment, partial observability, communication, and non-stationarity. Robustness and generalisation suites such as Melting Pot, Craftax-MA, HECOGrid, and ZSC-Eval extend evaluation beyond standard task performance by measuring adaptation to unseen partners, environments, and long-horizon scenarios. Finally, application-grounded benchmarks in robotics, autonomous driving, traffic control, energy systems, logistics, and sports increasingly align evaluation with domain-specific operational objectives, safety requirements, and deployment constraints.

A recurring theme across the benchmarking literature is the growing emphasis on evaluation quality itself. Recent work has highlighted the importance of robust statistical reporting, multi-seed evaluation, uncertainty quantification, cross-play analysis, and adversarial diagnostics. At the same time, several studies have demonstrated that inconsistent experimental protocols, weak baselines, limited reporting standards, and insufficient statistical analysis can substantially distort empirical conclusions. These findings suggest that improvements in benchmarking methodology are as important as advances in algorithm design for ensuring reliable scientific progress.

A broader observation emerging from the survey is that evaluation capabilities have expanded considerably faster than theoretical understanding. While benchmark ecosystems now provide increasingly diverse and realistic tests of coordination, adaptation, and scalability, theoretical guarantees regarding convergence, sample complexity, stability, and generalisation in MARL remain comparatively limited. Furthermore, although several benchmarks have begun exploring larger populations and more complex environments, convincing evidence of scalable coordination in substantially larger multi-agent systems remains scarce. Robustness to adversarial behaviour, distributional shift, and changing partner populations also continues to represent a major obstacle to reliable deployment in real-world settings.

Overall, MARL remains a rapidly evolving and highly promising research field. Future progress will depend on strengthening the connection between theoretical analysis and empirical evaluation, developing coordination mechanisms that generalise beyond narrow benchmark settings, and adopting standardised evaluation protocols that rigorously assess sample efficiency, robustness, scalability, and transferability. By consolidating recent developments in benchmarking and evaluation, examining their limitations, and identifying some most significant open challenges, this survey aims to provide a structured reference point and future research roadmap for the MARL community.

Author Contributions: Conceptualisation and methodology A.R. and M.E.A.; formal analysis A.R., K.P. and M.P.H.; writing—original draft preparation A.R., K.P. and M.P.H.; writing—review and editing, A.R., K.P., M.P.H. and M.E.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or generated in this study. Data sharing is not applicable to this article.

Acknowledgments: The authors used OpenAI's GPT-5.5 to assist in the preparation of illustrative figures. They have reviewed, edited, and validated all outputs.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper..

Appendix A. Multi-Agent Systems Overview

This section provides an overview of MAS, starting with their fundamental definition, key characteristics, and capabilities in decentralised problem solving. It then discusses how agents interact through cooperation, competition, and coordination, highlighting the mechanisms that enable effective collective behaviour in complex and dynamic environments. Finally, it presents key application domains of MAS, highlighting their impact across areas such as robotics, transportation, energy systems, cybersecurity, and healthcare.

Appendix A.1. Definition and Characteristics of MAS

A Multi-Agent System (MAS) consists of multiple interacting intelligent agents operating within a shared environment to achieve individual or collective goals [187]. An agent is commonly defined as an autonomous computational entity capable of perceiving its environment through sensors and acting upon it through actuators [188]. MAS research lies at the intersection of artificial intelligence (AI), distributed systems, and control theory, supporting decentralised problem solving in complex and dynamic environments [189].

Key characteristics of MAS include [188]: **Autonomy:** agents operate without direct human intervention and independently select actions according to internal decision policies; **Social Ability:** agents interact with others through communication protocols or shared environments; **Reactivity:** agents perceive environmental changes and respond in a timely manner; **Proactivity:** agents pursue long-term goals rather than acting only reactively; **Decentralisation:** decision-making is distributed across agents, improving robustness and reducing dependence on central control; and **Adaptability and Learning:** agents may improve behaviour over time through experience.

These characteristics provide the conceptual foundation for MARL. In MARL, autonomy is represented through agent-specific policies, social ability appears through strategic interactions and joint action spaces, reactivity and proactivity emerge through sequential decision-making, and adaptability is achieved through policy optimisation. Decentralisation further motivates settings where agents learn and act based on local observations rather than global system knowledge. To model these interactions formally, MARL commonly represents MAS as stochastic sequential decision problems [16]. At each time step, agents select actions jointly, causing transitions in the environment and receiving rewards that guide learning. Depending on the degree of observability and the alignment of objectives, different mathematical formulations are adopted, including stochastic (Markov) games and decentralised partially observable Markov decision processes (Dec-POMDPs) [22].

MASs are particularly suitable for large-scale and open environments where centralised control is impractical or inefficient [189]. Their distributed nature improves robustness, scalability, and flexibility when addressing complex computational problems.

Appendix A.2. Cooperation, Competition, and Coordination

Interactions among agents in MAS generally fall into three categories [33]: **Cooperation** occurs when agents pursue a common goal and share information or resources to maximise collective performance. Cooperative MAS commonly rely on task allocation, consensus mechanisms, and shared planning strategies [190]. In MARL, these settings are typically formulated as Dec-POMDPs, where agents operate under partial observations while optimising a shared reward function through a joint policy [22]. **Competition** arises when agents pursue conflicting goals or compete over limited resources. Competitive environments are frequently analysed using game-theoretic principles, where each agent adapts

to the behaviour of others [187]. In MARL, such settings are commonly represented as Stochastic games or Markov games, where each agent learns an individual policy and seeks to maximise its own expected return [16]. **Coordination and Mixed-Motive Interaction** refer to settings in which agents exhibit both aligned and conflicting goals [191]. Agents may cooperate within subgroups while competing against others or balance local rewards with system-level goals. These environments are frequently modelled in MARL using general-sum Markov games, where rewards are neither fully shared nor fully independent and learning requires coordination under mixed incentives [16,187].

In practical MAS deployments, these interaction modes often coexist. Agents may cooperate locally while competing globally, requiring learning approaches capable of balancing coordination, adaptation, and strategic behaviour across decentralised environments [189].

Appendix A.3. Applications of Multi-Agent Systems

MASs have been widely applied across diverse domains due to their capability to model distributed intelligence and complex interactions among autonomous entities [16,188,189]. In robotics and autonomous systems, MAS enable effective coordination among multiple robots for tasks such as exploration, search-and-rescue operations, and warehouse automation [2,3,5,13]. In smart grids and energy management, agents are used to manage distributed energy resources, optimise demand-response strategies, and enhance system resilience [17,169].

In transportation and traffic control, MAS-based approaches support intelligent traffic signal optimisation, autonomous vehicle coordination, and efficient route planning in smart city environments [1,166,168]. In cybersecurity, MAS facilitate distributed monitoring, intrusion detection, and adaptive defense mechanisms in dynamic and adversarial settings [17,21]. In e-commerce and market systems, autonomous agents enable automated negotiation, auction mechanisms, and supply chain coordination [187,189].

In healthcare systems, MAS contribute to patient monitoring, resource allocation, and decision support in distributed clinical environments [17,21]. The integration of MAS with machine learning (ML) and Internet-of-Things (IoT) technologies continues to expand their applicability in real-world cyber-physical systems, enabling scalable, adaptive, and intelligent infrastructures [8,12,16].

Overall, these developments highlight the growing importance of multi-agent systems as a unifying framework for modelling and solving complex, distributed problems. As advances in learning algorithms, communication strategies, and scalable architectures continue, MASs are expected to play an increasingly central role in next-generation intelligent systems.

Appendix B. Fundamentals of Deep Reinforcement Learning

Deep reinforcement learning is an interdisciplinary field that integrates RL with DL to enable agents to learn effective decision-making strategies directly from high-dimensional data [14]. This section presents the fundamental concepts required to understand DRL. It begins with classical RL principles, followed by a concise overview of DL, and then discusses how these paradigms are combined.

Appendix B.1. Reinforcement Learning Basics

Reinforcement learning is a branch of machine learning that focuses on how an autonomous agent can learn optimal behaviour through interactions with an environment. Unlike supervised learning, RL does not rely on labelled input-output pairs. Instead, the agent receives evaluative feedback in the form of rewards, which may be delayed or sparse, making the learning problem more challenging [192].

Appendix B.1.1. Markov Decision Processes

Reinforcement learning problems are commonly modelled as a Markov decision process (MDP) [192], defined by the tuple

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, R, \gamma) \quad (\text{A1})$$

where the following apply:

- $\mathcal{S}$ denotes the set of states describing the possible situations of the agent;
- $\mathcal{A}$ represents the set of actions available to the agent;
- $T(s_t, a_t, s_{t+1})$ is the state transition probability function, defining the dynamics of the environment;
- $R(s_t, a_t, s_{t+1})$ is the reward function, providing scalar feedback for taking action a_t in state s_t and transitioning to s_{t+1} ; and
- $\gamma \in [0, 1]$ is the discount factor, which controls the importance of future rewards.

The Markov property assumes that the future state depends only on the current state and action, not on the full history of interactions. This assumption simplifies the modelling of sequential decision-making problems and enables the use of efficient solution methods. Within this framework, the objective of the agent is to learn a policy that maximises the expected cumulative discounted reward over time, balancing short-term gains with long-term benefits

Appendix B.1.2. Policy and Value Functions

An agent's behaviour is characterised by a policy $\pi(a | s)$, which specifies the probability of selecting action a in state s . The objective of reinforcement learning is to find an optimal policy that maximises the expected cumulative discounted return

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad (\text{A2})$$

The **state-value function** and **action-value function** provide formal measures of how good states and state–action pairs are under a given policy π .

- **State-value function** $V^\pi(s)$ It is defined as the expected cumulative discounted return when starting from state s and following policy π thereafter:

$$V^\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$$

where G_t is the return. Intuitively, $V^\pi(s)$ measures how valuable it is for the agent to be in state s under policy π .

- **Action-value function** $Q^\pi(s, a)$ It is defined as the expected cumulative discounted return when taking action a in state s and then following policy π :

$$Q^\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$$

This function evaluates the quality of taking action a in state s under policy π .

These value functions form the foundation for many reinforcement learning algorithms, which iteratively estimate and refine them to improve the policy. Common approaches include dynamic programming, which relies on a known model of the environment; Monte Carlo methods, which use sampled episodes to estimate returns; and temporal-difference learning, which combines ideas from both by updating estimates based on partial experience. Together, these techniques enable agents to learn effective strategies

even in complex and uncertain environments. For a more detailed discussion readers are referred to [192].

Appendix B.1.3. Exploration and Exploitation

A central challenge in reinforcement learning is the exploration–exploitation trade-off. Exploration refers to trying new actions to discover potentially better strategies, while exploitation involves selecting actions known to yield high rewards. Balancing these two objectives is critical for effective learning, particularly in large or uncertain environments [192]. In practice, excessive exploration can lead to slow convergence and inefficient use of resources, whereas premature exploitation may cause the agent to converge to sub-optimal policies. Various strategies have been proposed to address this trade-off, including ϵ -greedy methods, entropy regularisation, upper confidence bound (UCB) approaches, and intrinsic motivation techniques that encourage agents to seek novel or informative states [31,192].

In multi-agent settings, this challenge becomes even more complex due to the presence of other learning agents. For example, value estimation becomes harder because the joint action space grows exponentially, and exploration becomes harder because other agents' policies change during learning. The environment appears non-stationary from each agent's perspective, as the policies of other agents evolve simultaneously [16,17,20]. This dynamic can make it difficult to distinguish whether changes in rewards are due to an agent's own actions or the behaviour of other agents, complicating exploration strategies [17,20].

Recent work in multi-agent deep reinforcement learning explores coordinated exploration, where agents share information or learn communication protocols to reduce redundant exploration and improve sample efficiency [7,23,56,70]. Moreover, techniques such as curiosity-driven learning, population-based training, and meta-learning have shown promise in enabling more adaptive and scalable exploration mechanisms in complex multi-agent environments [12,57,193].

Appendix B.2. Deep Learning Overview

Deep learning is a subfield of machine learning that focuses on learning hierarchical representations of data using artificial neural networks with multiple layers [194]. Its success in areas such as computer vision, speech recognition, and natural language processing arises from its ability to automatically extract relevant features from raw, high-dimensional inputs. Deep learning has advanced substantially over the past decades, transforming many fields through its ability to model highly complex patterns. Early deep learning architectures, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), established the foundation for many contemporary AI systems. CNNs, originally developed for image-related tasks, employ convolutional layers to learn spatial hierarchies within data, enabling state-of-the-art performance in applications such as image and video recognition [195]. In parallel, RNNs were specifically designed to model sequential data and have been widely applied in domains such as speech recognition and natural language processing [196]. However, early RNN architectures struggled to capture long-term dependencies due to the vanishing gradient problem. The introduction of Long Short-Term Memory (LSTM) units mitigated this limitation by enabling more stable gradient propagation, with substantially improving the learning of long-range temporal dependencies and overall model performance [26,194].

Appendix B.2.1. Neural Networks and Representation Learning

Artificial neural networks consist of layers of interconnected neurones, where each neurone computes a weighted sum of its inputs followed by a nonlinear activation function.

When multiple layers are stacked together, forming a deep neural network, the system can learn hierarchical representations, with early layers capturing low-level patterns and deeper layers recognising more abstract, high-level features. This layered learning is what gives deep networks their remarkable expressive power [196,197]. Several characteristics make deep learning particularly powerful [198,199], including (i) **End-to-end learning**, which reduces the need for manual feature engineering; (ii) **Scalability** to large datasets and complex tasks, especially with modern hardware accelerators; and (iii) **Generalisation** capabilities when trained with sufficient data and appropriate regularisation. In essence, deep learning combines flexible modelling, automated feature extraction, and computational efficiency, which together allow neural networks to tackle problems that were once considered extremely challenging.

Appendix B.2.2. Training Deep Models

Deep learning models are typically trained using gradient-based optimisation methods, most commonly variants of stochastic gradient descent, which iteratively update model parameters to minimise a predefined loss function. This optimisation process relies on backpropagation, an efficient algorithm that computes gradients by propagating error signals backward through the network layers. Together, these techniques enable deep neural networks to learn complex, non-linear mappings from data and have become the backbone of modern deep learning systems [194]. As multiple layers are stacked, networks are able to extract increasingly abstract representations, allowing them to perform effectively across a wide range of challenging tasks [195,197].

Despite their strong expressive power and flexibility, deep neural networks are not without limitations. One persistent challenge is overfitting, particularly when models contain a large number of parameters relative to the available training data. Without sufficient regularisation or data diversity, networks may learn spurious patterns that fail to generalise to unseen samples [194,195]. Moreover, training deep architectures can be hindered by vanishing or exploding gradients, especially in very deep or recurrent structures, which can slow convergence or destabilise learning [194,196]. Model performance is also highly sensitive to hyperparameter choices, such as learning rates, network depth, and optimisation settings, making training deep networks a careful and often computationally intensive process [17,195]. These challenges have motivated ongoing research into improved architectures, optimisation strategies, and regularisation techniques to ensure stable training and robust generalisation in practical applications [17,195].

Appendix B.3. Deep Reinforcement Learning: Combining RL and DL

Deep reinforcement learning combines the sequential decision-making framework of reinforcement learning with the representational power of deep learning [200]. In classical reinforcement learning, an agent interacts with an environment by observing states, taking actions, and receiving rewards, with the objective of learning a policy that maximises cumulative return over time. Traditional RL approaches typically depend on tabular value functions or carefully engineered features to represent the state or state–action space. While effective for small, discrete problems, these representations become infeasible as the dimensionality of the state space grows or when states and actions are continuous.

Deep neural networks overcome this limitation by acting as powerful, flexible function approximators capable of learning compact, high-level representations directly from raw or high-dimensional inputs, such as images, sensor readings, or continuous control signals [195]. By replacing tabular structures with neural networks, DRL enables the approximation of value functions, policies, or environment models in domains where explicit enumeration of states is impossible. This integration allows agents to generalise

across unseen states, capture complex nonlinear relationships, and scale to real-world problems [193].

Moreover, the use of deep learning has facilitated end-to-end training pipelines in which perception, representation learning, and decision-making are optimised jointly. Landmark algorithms such as Deep Q-Networks (DQN), policy gradient methods, and actor–critic architectures demonstrate how deep networks can be combined with reinforcement learning objectives to achieve human-level or superhuman performance in challenging domains [31,49]. As a result, DRL has emerged as a central paradigm for tackling large-scale, high-dimensional sequential decision-making problems that were previously out of reach for classical reinforcement learning methods.

Appendix B.4. Applications and Common DRL Architectures

In many real-world applications, such as robotics, autonomous driving, and game playing, agents must operate in environments characterised by high-dimensional sensory inputs and complex dynamics. These challenges lead to what is commonly known as the *curse of dimensionality*, which makes traditional reinforcement learning methods difficult to apply in complex, high-dimensional settings [31]. Deep reinforcement learning helps overcome this limitation by automatically learning compact, task-relevant representations directly from raw observations, reducing the need for manual feature engineering and enabling more scalable learning.

Deep neural networks in DRL can be used to approximate different components of the learning process: (i) **Value-based methods**, which approximate value or action-value functions; (ii) **Policy-based methods**, which directly learn a parameterised policy; and (iii) **Actor–critic methods**, which combine policy learning (actor) with value estimation (critic).

While powerful, DRL introduces new challenges, including training instability, non-stationary targets, and high variance in gradient estimates [49]. Various stabilisation techniques, such as experience replay and target networks, have been proposed to address these issues [31].

Appendix C. Benchmark Reporting Practices Across MARL Benchmark Suites

This appendix summarises reporting practices across the benchmark-suite papers discussed in the main text. Tables are split by benchmark category and kept to one page each where possible. Due to space constraints and the large number of benchmark papers, some details are omitted, for example, variations in evaluation protocols across algorithms reported in the same paper. The reader is referred to the original papers for full reporting details.

Table A1. Reporting practices in MARL evaluation-infrastructure papers.

Suite Paper	Algs/Baselines	Seeds	Tuning	Code	Metrics	Uncertainty	
PettingZoo [81]	Ape-X DQN, PPO	4/10	RL Baselines3 Zoo	Yes	Expected reward; avg. total reward	Shaded curves	
OpenSpiel [82]	CFR, XFP, ED, RCFR, A2C, NFSP, QPG, RM, RPG, NeuRD	5	Top-5 seeds/hp settings averaged	Yes	NashConv; exploitability; α -Rank	Shaded curves	
Standardised protocol [83]	IQL, VDN, QMIX demo	10	Demo untuned; protocol requires tuning disclosure	Yes	Return; win rate; IQM; opt. gap; perf. profiles	95% bootstrap CI	
Meta-game evaluation [84]	IDPPO, MAPPO, R-NaD, NFSP, PSRO, FCP, IS-MCTS, heuristics	10	LR tuned by SUMREGRET	NR	NE-Regret; Uniform; NE-NBS	95% CI; shaded curves	
EPyMARL [56]	IQL, IA2C, IPPO, MADDPG, COMA, MAA2C, MAPPO, VDN, QMIX	5; tune 3	Grid search on one task/env.	Yes	Max return; avg. return; SMAC win rate	95% CI	
BenchMARL [86]	MAPPO, IPPO, MADDPG, IDDPG, MASAC, ISAC, QMIX, VDN, IQL	3	Fine-tuned configs; protocol	NR	Yes	Norm. return; IQM; perf. profiles; opt. gap	95% bootstrap CI
JaxMARL [87]	IPPO, MAPPO, IQL, VDN, QMIX	10	Own tuning; exact protocol	NR	Yes	Norm. score; return; win rate; throughput	s.e.; 95% bootstrap CI

Abbreviations: CI, confidence interval; std., standard deviation; s.e., standard error; IQM, interquartile mean; perf., profiles performance profiles; opt. gap, optimality gap.

Table A2. Reporting practices in core-performance MARL benchmark papers.

Suite Paper	Algs/Baselines	Seeds	Tuning	Code	Metrics	Uncertainty
SMAC [88]	QMIX, COMA, VDN, IQL, QTRAN, heuristic	5	QTRAN hp search	Yes	Test win rate	Median; 25–75%
SMACv2 [89]	QMIX, MAPPO, IPPO, QPLEX, open-loop	3	Grid search	Yes	Mean test win rate; RMSE	std.
SMAC-Hard [90]	QMIX, QPLEX, LDSA, MAPPO, HAPPO	5	Official defaults	Yes	Win rate; expected return	Shaded variance
Hanabi [93]	ACHA, Rainbow, BAD; rule bots	50/30	PBT for ACHA	Yes	Mean score; perfect-game rate	std.; s.e.m.
Yokai [94]	SP, HE-IPPO, OP, OBL, human	12	Entropy/layer sweeps	Yes	Return; SEE; EE; WEE; length	Mean $\pm$ s.e.m.
OGC [95]	DR, PLR, ACCEL, PAIRED; MAPPO, FCP, MEP	5; eval 8	Grid search	Yes	Episode reward; solved rate	95% CI; s.e.; std.
OvercookedV2 [96]	PPO with SP, aug-SP, OP, FCP, BR	10	Explored arch.hp; protocol NR	Yes	SP; XP; SP–XP gap	Mean $\pm$ std.
MPE [24]	MADDPG, DDPG, DQN, REINFORCE, TRPO, AC	10/3	NR	Yes	Success; reward; collisions; distance	Shaded curves
MAgent [101]	DQN, DRQN, A2C	NR	NR	Yes	Reward; qualitative behaviour	NR

Abbreviations: SP, self-play; XP, cross-play; BR, best response; AC, actor–critic; HE, high-entropy.

Table A3. Reporting practices in robustness, generalisation, and specialised MARL benchmark papers.

Suite Paper	Algs/Baselines	Seeds	Tuning	Code	Metrics	Uncertainty
Off-the-Grid MARL [105]	BC, QMIX, QMIX+BCQ, QMIX+CQL, MAICQ; ITD3 variants; OMAR	10	Tune on representative tasks; reuse	Yes	Episode return; perf. profiles	std.; 95% bootstrap bands
Offline MARL audit [106]	BC, IQL+CQL, IDDPG+BC, MAD-DPG+CQL; MAICQ, OMAR, MADT, etc.	10	Fixed across scenarios	Yes	Episode return; norm. score; perf. profiles	std.; 95% CI t-test
ZSC-Eval [107]	FCP, MEP, TrajeDi, HSP, COLE, E3T, SP	5/3	Prior hp reused	Yes	BR-Prox; BR-Div; return; Spearman r_s	95% CI; IQR
Melting Pot [102]	A3C, V-MPO, OPRE; prosocial; exploiters; random	1/substrate	No tuning	Yes	Focal return; norm. score; Elo; equality	NR
Melting Pot 2.0 [108]	ACB, V-MPO, OPRE; prosocial; exploiters; random	3	Tuned on limited substrates	Yes	Focal return; norm. score; background return; inequality	std.
HECOGrid [110]	SAF, IPPO, MAPPO, QPLEX	NR	NR	Yes	Episodic return	Boxplots; shaded curves
Craftax-MA [111]	MAPPO, IPPO, PQN+VDN	3	Reused defaults	Yes	Reward; achievement/resource rate; trades	1 s.e.
MEAL [113]	IPPO+FT/L2/EWC/Online EWC/MAS/AGEM; MAPPO side study	5	Fixed settings	Yes	Avg. norm. score; forgetting; forward transfer; AUC-loss	95% CI; some std.
MOMAland [114]	MOMAPPO, scalarised IQL, GPI-LS, PCN	10/5	NR	Yes	Hypervolume; expected utility; cardinality; PF	95% CI
FightLadder [115]	PPO, IPPO, 2Timescale, FSP, PSRO, league	3	Fixed PPO hp	Yes	Win rate; Elo; exploitability	Mean $\pm$ std.
MADRID [85]	MADRID; targeted and random baselines	3; eval 4	Protocol NR	NR	Regret; scoring rate	s.e.
SocialJax [116]	IPPO-IR, IPPO-CR, MAPPO, VDN, SVO, IPPO-RE	NR	SVO grid sweep	Yes	Return; cooperation metrics; throughput	std.; shaded curves

Abbreviations: PF, Pareto front; FT, fine-tuning; perf., profiles performance profiles.

Table A4. Reporting practices in application-grounded MARL benchmark papers.

Suite Paper	Algs/Baselines	Seeds	Tuning	Code	Metrics	Uncertainty
Robotics, embodied collaboration, and sim-to-real						
Multi-Agent MuJoCo [117]	COMIX, FacMADDPG, MADDPG; ablations	10	No formal protocol	Yes	Mean episode return	95% CI
VMAS [118]	CPPO, MAPPO, IPPO, HetIPPO, heuristic	10	NR	Yes	Episode reward; exec. time	Mean $\pm$ std.
SMART [119]	IDQN, MADDPG, COMA, VDN, QMIX, MAAC	NR; 24 real eps	NR	Yes	Reward; collision; merge success; speed	NR
NeuronsMAE [120]	IQL, VDN, QMIX	NR	NR	Yes	Win rate	Shaded curves
IsaacLab-HARL [122]	HAPPO; alternating/simultaneous training	NR	NR	Yes	Win rate; reward	Shaded curves
Isaac Lab heterog. collab. [123]	MAPPO, HAPPO	3	MAPPO defaults	Yes	Avg. reward	std. shading
RoboFactory [124]	Diffusion Policy; 4 IL variants	100 scenes eval	NR	Yes	Success rate; ep. length	NR
PARTNR [125]	Heuristic, ReAct, ReAct+RAG, finetuned Llama, human teams	1000 eps	LoRA finetuning; val-set selection	Yes	Steps; success; completion; planning cycles	Mean $\pm$ s.e.
VIKI-R/VIKI-Bench [126]	GPT-4o, Claude, Gemini, Qwen-VL, Llama-Vision, ReAct, MAD, Ans-SFT, VIKI-R-Zero	NR	NR	Yes	Activation acc.; planning acc.; RMSE/HD/DFD	NR
MQE [127]	MAPPO, JRPO, MAT	5	NR	Yes	Step reward; FPS	std.; shaded variance
Assistax [128]	IPPO, MAPPO, MASAC; FF/RNN; meta-learning AHT	16; tune 12	168-config sweep; AUC selection	Yes	Return; norm. return; SPS	95% bootstrap CI
AT-Drone [129]	SP, PBT, MAPPO, HOLA-Drone, NAHT-D, BR	NR	NR	Yes	Success; collision; success steps; reward	Mean $\pm$ std.
MARBLER [130]	MAPPO, QMIX, VDN, QMIX_NS	5; 100 sim/10 real	Search on Simple Navigation; best-reward model	Yes	Reward; steps; left objects; collisions; OOB	95% CI; std.
JaxRobotarium [131]	PQN, QMIX, MAPPO, IPPO; MARBLER/RPS comparison	5; 3; 30	Width ablation; tuned defaults	Yes	Reward; task metrics; collisions; wall time	std.; s.e.
Path planning, warehouses, driving, and traffic						
POGEMA [134]	MAMBA, QPLEX, IQL, VDN, QMIX; DCC, SCRIMP; LaCAM, RHCR	1/10/128	Grid sweep	Yes	CSR; SoC; makespan; throughput	95% CI
CAMAR [135]	IPPO, MAPPO, IDDPG, MADDPG, ISAC, MASAC; RRT+PD, RRT*+PD, hybrids	3; hard 30	NR	Yes	Success; flowtime; makespan; coordination	IQM + 95% CI
GRACE [136]	CBSh2-RTC, EECBS, LaCAM, MAPF-LNS2, SCRIMP, MAPF-GPT-DDG, PSIPP/CTC, db-CBS	25/map	Default hp; official pre-trained models	Yes	Success; SoC; makespan; planning time; RTF	95% CI

Table A4. Cont.

Suite Paper	Algs/Baselines	Seeds	Tuning	Code	Metrics	Uncertainty
URB [137]	IQL, IPPO, MAPPO, QMIX; AON, Random, Human	5	Mentioned; protocol NR	Yes	Travel time; costs; speed/mileage change; WR	Mean $\pm$ std.; 95% CI
MRTA-Sim [138]	SMrTa, Nav2/A*, CBF-QP variants; Roadways; Room Queues	NR	NR	Yes	Control time; RTF	No
MAPF in 3D warehouses [139]	EECBS variants; MAPF-LNS2	10	Suboptimality comparison; add/subtract-one ablation	Dataset yes	Max solved; slowdown; success; norm. cost	NR
RWARE via EPyMARL [56]	IQL, IA2C, IPPO, MADDPG, COMA, MAA2C, MAPPO, VDN, QMIX	5; tune 3	Grid search	Yes	Max/avg. return	95% CI
TorchDriveEnv [143]	A2C, PPO, SAC, TD3	4; eval 10	NR	Yes	Collision; TL viol.; waypoint; off-road; return; horizon	Shaded curves
NAVSIM [144]	CV, Ego-MLP, TransFuser, LTF, UniAD, PARA-Drive, Hydra-MDP	3	NR	Yes	PDMS; NC; DAC; TTC; comfort; progress	std.
Bench2Drive [145]	UniAD, VAD, AD-MLP, TCP, ThinkTwice, DriveAdapter	NR	Official configs reused	Yes	DS; SR; efficiency; comfort; skill scores	No
GPUDrive [146]	IPPO; sim comparison vs Noc-turne/Waymax/MetaDrive	3	Tuned params reported; protocol NR	Yes	ASPS/CASPS; goal; collision; off-road; solve time	Shaded variance
MDrive [147]	TCP, UniAD, VAD, LMDrive; CoDriving, CoLM-Driver, CoLMDriver*	NR	NR	Yes	DS; SR; AP; ADE; collision rate	NR
IntersectionZoo [148]	PPO, DDPG, MAPPO, GCRL, IDM	4	RLlib defaults; best checkpoint selected	Yes	CO ₂ /vehicle; throughput	NR
PyTSC [149]	IQL, IA2C, VDN, QMIX, MAA2C; Fixed, Greedy, Max Pressure, SOTL	NR	Tune on 2x2/3x3, transfer after	Yes	Queue; delay; travel time	Mean and s.e.
VissimRL [150]	PPO; fixed-time; raw COM baseline	5	NR	Yes	Delay; travel time; waiting; throughput; speed; latency	Mean $\pm$ SD
Energy, infrastructure, sports, and space						
CommonPower [151]	PPO, IPPO, MAPPO; robust MPC; LSTM/Transformer	5	100 Ray Tune trials	Yes	Cost; failed seeds; interventions; safeguard time; RMSE	Min/max/mean; error bars
RL2Grid [152]	DQN, PPO, SAC, TD3, LagrPPO; heuristics; MILP/Idle	10; large 5	Grid search	Yes	Survival; line margins; overload penalty; topology changes	95% CI
EnEnv 1.0 [153]	A2C, PPO, DDPG, SAC, TD3; MAPPO, IPPO, MADDPG, IDDPG, MASAC, ISAC	5/3	Multiple net configs tested	Yes	Best eval reward	std.
MARLEM [154]	Zero-intelligence agents; PPO/APPO/SAC framework support	NR	NR	Yes	Welfare; liquidity; volatility; congestion; coordination	No

Table A4. Cont.

Suite Paper	Algs/Baselines	Seeds	Tuning	Code	Metrics	Uncertainty
MuFlex [155]	SAC; RBC	NR	Random search + OA design + grid search	Yes	Peak demand; reward; HVAC power; runtime	NR
OpenG2G [156]	NoCoord, Droop, OFO, PPO	5	Per-system tuning; best checkpoint by voltage violation	Yes	Voltage violation; tok/s; latency-target viol.; switching	Mean $\pm$ std.
Google Research Football [103]	IMPALA, PPO, Ape-X DQN	5; tune 3	100-config search	Yes	Avg. goal diff.	95% bootstrap CI; std.
RobocupGym [158]	PPO, SAC	NR	PPO settings reported; SAC defaults	Yes	Kick distance; ball velocity; reward; runtime	Shaded curves
MARLadona [159]	PPO trainee; Bot, HELIOS, self-play RL	NR	NR	Yes	Win/draw/loss; possession; passes; turnovers; duration	No
Space Robotics Bench [160]	PPO, TD3, DreamerV3; PPO-LSTM	3	Shared hp across tasks	Yes	Return; success; ATE; jerk; scooped volume; throughput	std.
OrbitZoo [161]	PPO, DDQN/DQN, DDPG	Eval 30	No exhaustive tuning; Bayes opt. for physics validation	Yes	MAPE; RMSE; collision prob.; miss dist.; fuel; reward	± 1 std. for scalability
SmallSatSim [162]	PD, MPCC, GP-MPCC, nonlinear MPC, PPO, VPG	Single-trial eval	NR	Yes	Lateral error; control effort; success; contact force; throughput	No
Omitted for insufficient verified reporting found: SOCIALGYM 2.0 [132], Arena [133], [140–142], [157]; Abbreviations: IL, imitation learning; OOB, out of bounds; AHT, ad hoc teamwork; CV, constant velocity; TL, traffic light; DS, driving score; SR, success rate; WR, win rate; RTF, real-time factor; NoCoord, no coordination; tok/s, tokens per second; OA, orthogonal array; ATE, average tracking error.						

References

- Chen, J.; Zhu, B.; Zhang, M.; Ling, X.; Ruan, X.; Deng, Y.; Guo, N. Multi-Agent Deep Reinforcement Learning Cooperative Control Model for Autonomous Vehicle Merging into Platoon in Highway. *World Electr. Veh. J.* **2025**, *16*, 225.
- Dong, P.; Liu, J.; Tao, H.; Zhao, Y.; Feng, Z.; Luo, H. Long-Endurance Collaborative Search and Rescue Based on Maritime Unmanned Systems and Deep-Reinforcement Learning. *Sensors* **2025**, *25*, 4025.
- Lv, Y.; Lei, J.; Yi, P. A Local Information Aggregation-Based Multiagent Reinforcement Learning for Robot Swarm Dynamic Task Allocation. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 10437–10449.
- Zhao, B.; Qin, D.; Chen, Y.; Yang, J.; Tang, H.; Ma, L. Multi-agent deep reinforcement learning for RIS-assisted secure UAV communication. *J. King Saud Univ.-Comput. Inf. Sci.* **2025**, *37*, 133. <https://doi.org/10.1007/s44443-025-00145-9>.
- Calzolari, G.; Sumathy, V.; Kanellakis, C.; Nikolakopoulos, G. Reinforcement Learning Driven Multi-Robot Exploration via Explicit Communication and Density-Based Frontier Search. In Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA), Atlanta, GA, USA, 19–23 May 2025; pp. 11406–11412. <https://doi.org/10.1109/ICRA55743.2025.11128566>.
- Li, H.; Yang, P.; Liu, W.; Yan, S.; Zhang, X.; Zhu, D. Multi-Agent Reinforcement Learning in Games: Research and Applications. *Biomimetics* **2025**, *10*, 375. <https://doi.org/10.3390/biomimetics10060375>.
- Zhu, C.; Dastani, M.; Wang, S. A survey of multi-agent deep reinforcement learning with communication. *Auton. Agents Multi-Agent Syst.* **2024**, *38*, 4. <https://doi.org/10.1007/s10458-023-09633-6>.
- Ning, Z.; Xie, L. A survey on multi-agent reinforcement learning and its application. *J. Autom. Intell.* **2024**, *3*, 73–91.
- Huh, D.; Mohapatra, P. Multi-agent Reinforcement Learning: A Comprehensive Survey. *arXiv* **2024**, arXiv:2312.10256.
- Sun, W.-F.; Lee, C.-K.; See, S.; Lee, C.-Y. A Unified Framework for Factorizing Distributional Value Functions for Multi-Agent Reinforcement Learning. *J. Mach. Learn. Res. (JMLR)* **2023**, *24*, 1–32.
- Oh, J.; Farquhar, G.; Kemaev, I.; Calian, D.A.; Hessel, M.; Zintgraf, L.; Singh, S.; van Hasselt, H.; Silver, D. Discovering state-of-the-art reinforcement learning algorithms. *Nature* **2025**, *648*, 312–319. <https://doi.org/10.1038/s41586-025-09761-x>.
- Liang, J.; Miao, H.; Li, K.; Tan, J.; Wang, X.; Luo, R.; Jiang, Y. A Review of Multi-Agent Reinforcement Learning Algorithms. *Electronics* **2025**, *14*, 820. <https://doi.org/10.3390/electronics14040820>.
- Tang, C.; Abbatematteo, B.; Hu, J.; Chandra, R.; Martín-Martín, R.; Stone, P. Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, 25 February–4 March 2025; Volume 39, pp. 28694–28698.
- Terven, J. Deep Reinforcement Learning: A Chronological Overview and Methods. *AI* **2025**, *6*, 46. <https://doi.org/10.3390/ai6030046>.
- Dignum, F.; Mathieu, P.; Corchado, J.; De la Prieta, F. Advances in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation. In Proceedings of the PAAMS Collection: 20th International Conference, PAAMS 2022, L'Aquila, Italy, 13–15 July 2022.
- Zhang, K.; Yang, Z.; Başar, T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In *Handbook of Reinforcement Learning and Control*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 321–384.
- Nguyen, T.T.; Nguyen, N.D.; Nahavandi, S. Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Trans. Cybern.* **2020**, *50*, 3826–3839.
- Everett, M.; Chen, Y.F.; How, J.P. Collision Avoidance in Pedestrian-Rich Environments with Deep Reinforcement Learning. *IEEE Access* **2021**, *9*, 10357–10377.
- Shen, G.; Lei, L.; Zhang, X.; Li, Z.; Cai, S.; Zhang, L. Multi-UAV Cooperative Search Based on Reinforcement Learning with a Digital Twin Driven Training Framework. *IEEE Trans. Veh. Technol.* **2023**, *72*, 8354–8368.
- Hernandez-Leal, P.; Kartal, B.; Taylor, M.E. A survey and critique of multiagent deep reinforcement learning. *Auton. Agents Multi-Agent Syst.* **2019**, *33*, 750–797.
- Gronauer, S.; Diepold, K. Multi-agent Deep Reinforcement Learning: A Survey. *Artif. Intell. Rev.* **2022**, *55*, 895–943.
- Oliehoek, F.A.; Amato, C. *A Concise Introduction to Decentralized POMDPs*; Springer: Berlin/Heidelberg, Germany, 2016.
- Foerster, J.; Assael, I.A.; De Freitas, N.; Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 2145–2153.
- Lowe, R.; Wu, Y.I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6382–6393.
- Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; Whiteson, S. Counterfactual multi-agent policy gradients. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
- Graves, A. Long short-term memory. In *Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 37–45.
- Cho, K.; Van Merriënboer, B.; Gulçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1724–1734.

28. Hausknecht, M.J.; Stone, P. Deep Recurrent Q-Learning for Partially Observable MDPs. In Proceedings of the AAAI Fall Symposia, Arlington, VA, USA, 12–14 November 2015; Volume 45, p. 141.
29. Claus, C.; Boutilier, C. The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI/IAAI), Madison, WI, USA, 26–30 July 1998; pp. 746–752.
30. Tan, M. Multi-agent reinforcement learning: Independent versus cooperative agents. In *Proceedings of the Tenth International Conference on Machine Learning (ICML)*; Morgan Kaufmann: San Francisco, CA, USA, 1993; pp. 330–337.
31. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.A.; Fidjeland, A.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533.
32. Watkins, C.J.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292.
33. Busoniu, L.; Babuska, R.; De Schutter, B. A comprehensive survey of multiagent reinforcement learning. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2008**, *38*, 156–172.
34. Matignon, L.; Laurent, G.J.; Le Fort-Piat, N. Independent reinforcement learners in cooperative Markov games: A survey regarding coordination problems. *Knowl. Eng. Rev.* **2012**, *27*, 1–31.
35. Matignon, L.; Laurent, G.J.; Le Fort-Piat, N. Hysteretic Q-learning: An algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, USA, 29 October–2 November 2007; pp. 64–69.
36. van Hasselt, H.; Guez, A.; Silver, D. Deep reinforcement learning with double Q-learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
37. Wang, Z.; Schaul, T.; Hessel, M.; van Hasselt, H.; Lanctot, M.; de Freitas, N. Dueling Network Architectures for Deep Reinforcement Learning. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 20–22 June 2016; pp. 1995–2003.
38. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized Experience Replay. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, PR, USA, 2–4 May 2016.
39. Hessel, M.; Modayil, J.; van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, M.G.; Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
40. Bellemare, M.G.; Dabney, W.; Munos, R. A Distributional Perspective on Reinforcement Learning. In Proceedings of the 34th International Conference on Machine Learning (ICML), PMLR, Sydney, Australia, 6–11 August 2017; pp. 449–458.
41. Dabney, W.; Rowland, M.; Bellemare, M.G.; Munos, R. Distributional Reinforcement Learning with Quantile Regression. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI), New Orleans, LA, USA, 2–7 February 2018; pp. 2892–2901.
42. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.
43. van Hasselt, H.; Doron, Y.; Strub, F.; Hessel, M.; Sonnerat, N.; Modayil, J. Deep Reinforcement Learning and the Deadly Triad. *arXiv* **2018**, arXiv:1812.02648.
44. Son, K.; Kim, D.; Kang, W.J.; Hostallero, D.E.; Yi, Y. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 5887–5896.
45. Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W.M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J.Z.; Tuyls, K.; et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv* **2017**, arXiv:1706.05296.
46. Rashid, T.; Samvelyan, M.; De Witt, C.S.; Farquhar, G.; Foerster, J.; Whiteson, S. Monotonic value function factorisation for deep multi-agent reinforcement learning. *J. Mach. Learn. Res.* **2020**, *21*, 1–51.
47. Wang, J.; Ren, Z.; Han, B.; Ye, J.; Zhang, C. Towards understanding cooperative multi-agent q-learning with value factorization. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 29142–29155.
48. Wang, J.; Ren, Z.; Liu, T.; Yu, Y.; Zhang, C. Qplex: Duplex dueling multi-agent q-learning. *arXiv* **2020**, arXiv:2008.01062.
49. Sutton, R.S.; McAllester, D.; Singh, S.; Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 13th International Conference on Neural Information Processing Systems (NIPS’99)*; MIT Press: Cambridge, MA, USA, 1999; pp. 1057–1063.
50. Williams, R.J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **1992**, *8*, 229–256.
51. Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.I.; Moritz, P. Trust Region Policy Optimization. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 6–11 July 2015; pp. 1889–1897.
52. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.P.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 20–22 June 2016; pp. 1928–1937.

53. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
54. Yu, C.; Velu, A.; Vinitzky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The surprising effectiveness of PPO in cooperative multi-agent games. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24611–24624.
55. Engstrom, L.; Ilyas, A.; Santurkar, S.; Tsipras, D.; Janoos, F.; Rudolph, L.; Madry, A. Implementation Matters in Deep RL: A Case Study on PPO and TRPO. In Proceedings of the International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020.
56. Papoudakis, G.; Christianos, F.; Schäfer, L.; Albrecht, S.V. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative Tasks. In Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS), Virtual, 6–14 December 2021.
57. Moerland, T.M.; Broekens, J.; Plaat, A.; Jonker, C.M. Model-based Reinforcement Learning: A Survey. *Found. Trends Mach. Learn.* **2023**, *16*, 1–118.
58. Sutton, R.S. Dyna, an Integrated Architecture for Learning, Planning, and Reacting. *ACM SIGART Bull.* **1991**, *2*, 160–163.
59. Janner, M.; Fu, J.; Zhang, M.; Levine, S. When to Trust Your Model: Model-Based Policy Optimization. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
60. Chua, K.; Calandra, R.; McAllister, R.; Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 4754–4765.
61. Coulom, R. Efficient selectivity and backup operators in Monte-Carlo tree search. In *Proceedings of the International Conference on Computers and Games*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 72–83.
62. Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; et al. A General Reinforcement Learning Algorithm that Masters Chess, Shogi, and Go through Self-play. *Science* **2018**, *362*, 1140–1144.
63. Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature* **2020**, *588*, 604–609.
64. Hafner, D.; Lillicrap, T.; Ba, J.; Norouzi, M. Dream to Control: Learning Behaviors by Latent Imagination. *arXiv* **2020**, arXiv:1912.01603.
65. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic Policy Gradient Algorithms. In Proceedings of the International Conference on Machine Learning, PMLR, Beijing, China, 22–24 June 2014; pp. 387–395.
66. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
67. Iqbal, S.; Sha, F. Actor-attention-critic for multi-agent reinforcement learning. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 2961–2970.
68. Peng, B.; Rashid, T.; Schroeder de Witt, C.; Kamienny, P.A.; Torr, P.; Böhmer, W.; Whiteson, S. FACMAC: Factored Multi-Agent Centralised Policy Gradients. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 12208–12221.
69. Kuba, J.G.; Chen, R.; Wen, M.; Wen, Y.; Sun, F.; Wang, J.; Yang, Y. Trust region policy optimisation in multi-agent reinforcement learning. In Proceedings of the International Conference on Learning Representations, Virtual, 25–29 April 2022.
70. Sukhbaatar, S.; Szlam, A.; Fergus, R. Learning multiagent communication with backpropagation. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 4754–4765.
71. Das, A.; Gervet, T.; Romoff, J.; Batra, D.; Parikh, D.; Rabbat, M.; Pineau, J. Tarmac: Targeted multi-agent communication. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 1538–1546.
72. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Networks* **2009**, *20*, 61–80.
73. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
74. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
75. Battaglia, P.W.; Hamrick, J.B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; et al. Relational Inductive Biases, Deep Learning, and Graph Networks. *arXiv* **2018**, arXiv:1806.01261.
76. Jiang, J.; Dun, C.; Huang, T.; Lu, Z. Graph Convolutional Reinforcement Learning. In Proceedings of the International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020.
77. Guestrin, C.; Lagoudakis, M.; Parr, R. Coordinated Reinforcement Learning. In *Proceedings of the Nineteenth International Conference on Machine Learning (ICML)*; Morgan Kaufmann: San Francisco, CA, USA, 2002; pp. 227–234.
78. Böhmer, W.; Kurin, V.; Whiteson, S. Deep Coordination Graphs. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 980–991.

79. Liu, Y.; Wang, W.; Hu, Y.; Hao, J.; Chen, X.; Gao, Y. Multi-Agent Game Abstraction via Graph Attention Neural Network. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 7211–7218.
80. Li, S.; Zhou, Y.; Allen, R.; Kochenderfer, M.J. Deep Implicit Coordination Graphs for Multi-Agent Reinforcement Learning. In Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), Virtual, 3–7 May 2021; pp. 764–772.
81. Terry, J.K.; Black, B.; Grammel, N.; Jayakumar, M.; Hari, A.; Sullivan, R.; Santos, L.; Pérez, R.; Horsch, C.; Dieffendahl, C.; et al. PettingZoo: Gym for Multi-Agent Reinforcement Learning. *arXiv* **2020**, arXiv:2009.14471.
82. Lanctot, M.; Lockhart, E.; Lespiau, J.B.; Zambaldi, V.; Upadhyay, S.; Pérolat, J.; Srinivasan, S.; Timbers, F.; Tuyls, K.; Omidshafiei, S.; et al. OpenSpiel: A framework for reinforcement learning in games. *arXiv* **2019**, arXiv:1908.09453.
83. Gorsane, R.; Mahjoub, O.; de Kock, R.; Dubb, R.; Singh, S.S.; Pretorius, A. Towards a Standardised Performance Evaluation Protocol for Cooperative MARL. *arXiv* **2022**, arXiv:2209.10485.
84. Li, Z.; Wellman, M.P. A meta-game evaluation framework for deep multiagent reinforcement learning. *arXiv* **2024**, arXiv:2405.00243.
85. Samvelyan, M.; Paglieri, D.; Jiang, M.; Parker-Holder, J.; Rocktäschel, T. Multi-Agent Diagnostics for Robustness via Illuminated Diversity. *arXiv* **2024**, arXiv:2401.13460.
86. Bettini, M.; Prorok, A.; Moens, V. Benchmark: Benchmarking multi-agent reinforcement learning. *J. Mach. Learn. Res.* **2024**, *25*, 1–10.
87. Rutherford, A.; Ellis, B.; Gallici, M.; Cook, J.; Lupu, A.; Ingvarsson, G.; Willi, T.; Hammond, R.; Khan, A.; de Witt, C.S.; et al. Jaxmarl: Multi-agent rl environments and algorithms in jax. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 50925–50951.
88. Samvelyan, M.; Rashid, T.; De Witt, C.S.; Farquhar, G.; Nardelli, N.; Rudner, T.G.; Hung, C.M.; Torr, P.H.; Foerster, J.; Whiteson, S. The starcraft multi-agent challenge. *arXiv* **2019**, arXiv:1902.04043.
89. Ellis, B.; Cook, J.; Moalla, S.; Samvelyan, M.; Sun, M.; Mahajan, A.; Foerster, J.N.; Whiteson, S. SMACv2: An Improved Benchmark for Cooperative Multi-Agent Reinforcement Learning. *arXiv* **2023**, arXiv:2212.07489.
90. Deng, Y.; Yu, Y.; Ma, W.; Wang, Z.; Zhu, W.; Zhao, J.; Zhang, Y. SMAC-hard: Enabling mixed opponent strategy script and self-play on SMAC. *arXiv* **2024**, arXiv:2412.17707.
91. Wang, Y.; Han, B.; Wang, T.; Dong, H.; Zhang, C. DOP: Off-Policy Multi-Agent Decomposed Policy Gradients. *arXiv* **2020**, arXiv:2007.12322.
92. Geng, M.; Pateria, S.; Subagdja, B.; Tan, A.H. MOSMAC: A Multi-agent Reinforcement Learning Benchmark on Sequential Multi-Objective Tasks. In Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '25, Detroit, Michigan, USA, 19–23 May 2025; International Foundation for Autonomous Agents and Multiagent Systems: Richland, SC, USA, 2025; pp. 867–876.
93. Bard, N.; Foerster, J.N.; Chandar, S.; Burch, N.; Lanctot, M.; Song, H.F.; Parisotto, E.; Dumoulin, V.; Moitra, S.; Hughes, E.; et al. The hanabi challenge: A new frontier for ai research. *Artif. Intell.* **2020**, *280*, 103216.
94. Ruhdorfer, C.; Bortoletto, M.; Forkel, J.; Foerster, J.; Bulling, A. The Yokai Learning Environment: Tracking Beliefs Over Space and Time. *arXiv* **2026**, arXiv:2508.12480.
95. Ruhdorfer, C.; Bortoletto, M.; Penzkofer, A.; Bulling, A. The Overcooked Generalisation Challenge: Evaluating Cooperation with Novel Partners in Unknown Environments Using Unsupervised Environment Design. *arXiv* **2025**, arXiv:2406.17949.
96. Gessler, T.; Dizdarevic, T.; Calinescu, A.; Ellis, B.; Lupu, A.; Foerster, J. Overcookedv2: Rethinking overcooked for zero-shot coordination. In Proceedings of the International Conference on Learning Representations, Singapore, 24–28 April 2025; Volume 2025, pp. 62752–62784.
97. Roy, J.; Barde, P.; Harvey, F.G.; Nowrouzezahrai, D.; Pal, C. Promoting Coordination through Policy Regularization in Multi-Agent Deep Reinforcement Learning. *arXiv* **2020**, arXiv:1908.02269.
98. Li, Z.; Zhong, H.; Wang, X.; Xia, Q.; Zhang, L.; Huang, L. Diffusing to Coordinate: Efficient Online Multi-Agent Diffusion Policies. *arXiv* **2026**, arXiv:2602.18291.
99. Wu, H.; Song, S.; Yao, C.; Han, S.; Wan, H.; Lin, Y.; Lv, K. Think How Your Teammates Think: Active Inference Can Benefit Decentralized Execution. *arXiv* **2025**, arXiv:2511.18761.
100. Kilinc, O.; Montana, G. Multi-agent Deep Reinforcement Learning with Extremely Noisy Observations. *arXiv* **2018**, arXiv:1812.00922.
101. Zheng, L.; Yang, J.; Cai, H.; Zhang, W.; Wang, J.; Yu, Y. MAgent: A Many-Agent Reinforcement Learning Platform for Artificial Collective Intelligence. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018. <https://doi.org/10.1609/aaai.v32i1.11371>.
102. Leibo, J.Z.; Duéñez-Guzmán, E.; Vezhnevets, A.S.; Agapiou, J.P.; Sunehag, P.; Koster, R.; Matyas, J.; Beattie, C.; Mordatch, I.; Graepel, T. Scalable Evaluation of Multi-Agent Reinforcement Learning with Melting Pot. *arXiv* **2021**, arXiv:2107.06857.

103. Kurach, K.; Raichuk, A.; Stańczyk, P.; Zajac, M.; Bachem, O.; Espeholt, L.; Riquelme, C.; Vincent, D.; Michalski, M.; Bousquet, O.; et al. Google Research Football: A Novel Reinforcement Learning Environment. *arXiv* **2019**, arXiv:1907.11180.
104. Yang, Y.; Ma, X.; Li, C.; Zheng, Z.; Zhang, Q.; Huang, G.; Yang, J.; Zhao, Q. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 10299–10312.
105. Formanek, C.; Jeewa, A.; Shock, J.; Pretorius, A. Off-the-Grid MARL: Datasets with Baselines for Offline Multi-Agent Reinforcement Learning. *arXiv* **2023**, arXiv:2302.00521.
106. Formanek, C.; Tilbury, C.R.; Beyers, L.; Shock, J.; Pretorius, A. Dispelling the mirage of progress in offline marl through standardised baselines and evaluation. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 139650–139672.
107. Wang, X.; Zhang, S.; Zhang, W.; Dong, W.; Chen, J.; Wen, Y.; Zhang, W. Zsc-eval: An evaluation toolkit and benchmark for multi-agent zero-shot coordination. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 47344–47377.
108. Agapiou, J.; Vezhnevets, A.; Duéñez-Guzmán, E.A.; Matyas, J.; Mao, Y.; Sunehag, P.; Koster, R.; Madhushani, U.; Kopparapu, K.; Comanescu, R.; et al. Melting Pot 2.0. *arXiv* **2022**, arXiv:2211.13746.
109. Ndousse, K.K.; Eck, D.; Levine, S.; Jaques, N. Emergent social learning via multi-agent reinforcement learning. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 7991–8004.
110. Liu, D.; Shah, V.; Boussif, O.; Meo, C.; Goyal, A.; Shu, T.; Mozer, M.; Heess, N.; Bengio, Y. Stateful active facilitator: Coordination and Environmental Heterogeneity in Cooperative Multi-Agent Reinforcement Learning. *arXiv* **2022**, arXiv:2210.03022.
111. Omari, B.A.; Matthews, M.; Rutherford, A.; Foerster, J. Multi-Agent Craftax: Benchmarking Open-Ended Multi-Agent Reinforcement Learning at the Hyperscale. *arXiv* **2025**, arXiv:2511.04904.
112. Matthews, M.; Beukman, M.; Ellis, B.; Samvelyan, M.; Jackson, M.; Coward, S.; Foerster, J. Craftax: A lightning-fast benchmark for open-ended reinforcement learning. *arXiv* **2024**, arXiv:2402.16801.
113. Tomilin, T.; van den Boogaard, L.; Garcin, S.; Grooten, B.; Fang, M.; Du, Y.; Pechenizkiy, M. MEAL: A Benchmark for Continual Multi-Agent Reinforcement Learning. *arXiv* **2025**, arXiv:2506.14990.
114. Felten, F.; Ucak, U.; Azmani, H.; Peng, G.; Röpke, W.; Baier, H.; Mannion, P.; Roijers, D.M.; Terry, J.K.; Talbi, E.G.; et al. Momaland: A set of benchmarks for multi-objective multi-agent reinforcement learning. *arXiv* **2024**, arXiv:2407.16312.
115. Li, W.; Ding, Z.; Karten, S.; Jin, C. FightLadder: A benchmark for competitive multi-agent reinforcement learning. *arXiv* **2024**, arXiv:2406.02081.
116. Guo, Z.; Willis, R.; Shi, S.; Tomilin, T.; Leibo, J.Z.; Du, Y. SocialJax: An Evaluation Suite for Multi-agent Reinforcement Learning in Sequential Social Dilemmas. *arXiv* **2025**, arXiv:2503.14576.
117. de Witt, C.S.; Peng, B.; Kamienny, P.-A.; Torr, P.; Bohmer, W.; Whiteson, S. Deep Multi-Agent Reinforcement Learning for Decentralized Continuous Cooperative Control. *arXiv* **2020**, arXiv:2003.06709v2.
118. Bettini, M.; Kortvelesy, R.; Blumenkamp, J.; Prorok, A. VMAS: A Vectorized Multi-Agent Simulator for Collective Robot Learning. In Proceedings of the International Symposium on Distributed Autonomous Robotic Systems, Montbéliard, France, 28–30 November 2022; pp. 42–56.
119. Liang, Z.; Cao, J.; Jiang, S.; Saxena, D.; Chen, J.; Xu, H. From Multi-agent to Multi-robot: A Scalable Training and Evaluation Platform for Multi-robot Reinforcement Learning. *arXiv* **2022**, arXiv:2206.09590.
120. Hu, G.; Li, H.; Liu, S.; Ma, M.; Zhu, Y.; Zhao, D. NeuronsMAE: A Novel Multi-Agent Reinforcement Learning Environment for Cooperative and Competitive Multi-Robot Tasks. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 18–23 June 2023. <https://doi.org/10.1109/IJCNN54540.2023.10191291>.
121. Mittal, M.; Roth, P.; Tigue, J.; Richard, A.; Zhang, O.; Du, P.; Serrano-Munoz, A.; Yao, X.; Zurbrugg, R.; Rudin, N.; et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv* **2025**, arXiv:2511.04831.
122. Peterson, I.; Allred, C.; Morrey, J.; Harper, M. A Framework for Scalable Heterogeneous Multi-Agent Adversarial Reinforcement Learning in IsaacLab. *arXiv* **2025**, arXiv:2510.01264.
123. Haight, J.; Peterson, I.; Allred, C.; Harper, M. Heterogeneous Multi-Agent Learning in Isaac Lab: Scalable Simulation for Robotic Collaboration. In Proceedings of the 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hangzhou, China, 19–25 October 2025; pp. 13446–13451.
124. Qin, Y.; Kang, L.; Song, X.; Yin, Z.; Liu, X.; Liu, X.; Zhang, R.; Bai, L. RoboFactory: Exploring Embodied Agent Collaboration with Compositional Constraints. In Proceedings of the 2025 IEEE/CVF International Conference on Computer Vision (ICCV), Honolulu, HI, USA, 19–23 October 2025; pp. 10075–10085.
125. Chang, M.; Chhablani, G.; Clegg, A.; Cote, M.D.; Desai, R.; Hlavac, M.; Karashchuk, V.; Krantz, J.; Mottaghi, R.; Parashar, P.; et al. PARTNR: A Benchmark for Planning and Reasoning in Embodied Multi-agent Tasks. In Proceedings of the International Conference on Learning Representations, Singapore, 24–28 April 2025.
126. Kang, L.; Song, X.; Zhou, H.; Qin, Y.; Yang, J.; Liu, X.; Torr, P.; Bai, L.; Yin, Z. VIKI-R: Coordinating Embodied Multi-Agent Cooperation via Reinforcement Learning. *arXiv* **2025**, arXiv:2506.09049.

127. Xiong, Z.; Chen, B.; Huang, S.; Tu, W.-W.; He, Z.; Gao, Y. MQE: Unleashing the Power of Interaction with Multi-agent Quadruped Environment. In Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 14–18 October 2024; pp. 5918–5924.
128. Hinckeldey, L.; Fosong, E.; Rubavicius, R.; Miller, E.; McInroe, T.; Zhang, F.; Wollstadt, P.; Albrecht, S.V.; Ramamoorthy, S. Assistax: A Multi-Agent Hardware-Accelerated Reinforcement Learning Benchmark for Assistive Robotics. *arXiv* **2025**, arXiv:2507.21638.
129. Li, Y.; Chen, J.; Xue, F.; Qiu, J.; Li, W.; Zhang, Q.; Wen, Y.; Pan, W. AT-Drone: Benchmarking Adaptive Teaming in Multi-Drone Pursuit. *arXiv* **2025**, arXiv:2502.09762.
130. Torbati, R.; Lohiya, S.; Singh, S.; Nigam, M.S.; Ravichandar, H. MARBLER: An Open Platform for Standardized Evaluation of Multi-Robot Reinforcement Learning Algorithms. In Proceedings of the 2023 International Symposium on Multi-Robot and Multi-Agent Systems (MRS), Boston, MA, USA, 4–5 December 2023. <https://doi.org/10.1109/MRS60187.2023.10416792>.
131. Jain, S.; Liu, J.; Kailas, S.; Ravichandar, H. JaxRobotarium: Training and Deploying Multi-Robot Policies in 10 Minutes. *arXiv* **2025**, arXiv:2505.06771.
132. Sprague, Z.; Chandra, R.; Holtz, J.; Biswas, J. SOCIALGYM 2.0: Simulator for Multi-Agent Social Robot Navigation in Shared Human Spaces. *arXiv* **2023**, arXiv:2303.05584.
133. Shcherbyna, V.; Kästner, L.; Diaz, D.; Nguyen, H.G.; Schreff, M.H.-K.; Lenz, T.; Kreutz, J.; Martban, A.; Zeng, H.; Soh, H. Arena 4.0: A Comprehensive Ros2 Development and Benchmarking Platform for Human-Centric Navigation Using Generative-Model-Based Environment Generation. In Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA), Atlanta, GA, USA, 19–23 May 2025; pp. 9138–9144.
134. Skrynnik, A.; Andreychuk, A.; Borzilov, A.; Chernyavskiy, A.; Yakovlev, K.; Panov, A. POGEMA: A Benchmark Platform for Cooperative Multi-Agent Pathfinding. *arXiv* **2025**, arXiv:2407.14931.
135. Pshenitsyn, A.; Panov, A.; Skrynnik, A. CAMAR: Continuous Actions Multi-Agent Routing. *arXiv* **2025**, arXiv:2508.12845.
136. Zang, C.; Mannucci, A.; Barz, I.; Schillinger, P.; Lier, F.; Hönig, W. GRACE: A Unified 2D Multi-Robot Path Planning Simulator&Benchmark for Grid, Roadmap, And Continuous Environments. *arXiv* **2026**, arXiv:2603.10858.
137. Akman, A.O.; Psarou, A.; Hoffmann, M.; Gorczyca, Ł.; Kowalski, Ł.; Gora, P.; Jamróz, G.; Kucharski, R. URB–Urban Routing Benchmark for RL-equipped Connected Autonomous Vehicles. *arXiv* **2025**, arXiv:2505.17734.
138. Tuck, V.M.; Parwana, H.; Chen, P.-W.; Fainekos, G.; Hoxha, B.; Okamoto, H.; Sastry, S.S.; Seshia, S.A. MRTA-Sim: A Modular Simulator for Multi-Robot Allocation, Planning, and Control in Open-World Environments. *arXiv* **2025**, arXiv:2504.15418.
139. Wang, Q.; Veerapaneni, R.; Wu, Y.; Li, J.; Likhachev, M. MAPF in 3D Warehouses: Dataset and Analysis. In Proceedings of the International Conference on Automated Planning and Scheduling, Banff, AB, Canada, 1–6 June 2024; pp. 623–632.
140. Teixeira, A.; Moura, A. Realistic Multi-Agent Path Planning in Constrained Industrial Environments Applied to New Task Execution Unit Benchmarks. In Proceedings of the 15th International Conference on Operations Research and Enterprise Systems, Marbella, Spain, 9–11 March 2026.
141. Nagai, H.; Okumura, K. From Gridworlds to Warehouses: Adapting Lightweight One-shot Multi-Agent Pathfinding for AGVs. *arXiv* **2026**, arXiv:2605.15799.
142. Jiang, Y.; Zhan, G.; Lan, Z.; Liu, C.; Cheng, B.; Li, S.E. A Reinforcement Learning Benchmark for Autonomous Driving in General Urban Scenarios. In *IEEE Transactions on Intelligent Transportation Systems*; IEEE: New York, NY, USA, 2024; pp. 4335–4345.
143. Lavington, J.W.; Zhang, K.; Lioutas, V.; Niedoba, M.; Liu, Y.; Green, D.; Naderiparizi, S.; Liang, X.; Dabiri, S.; Ścibior, A.; et al. TorchDriveEnv: A Reinforcement Learning Benchmark for Autonomous Driving with Reactive, Realistic, and Diverse Non-Playable Characters. *arXiv* **2024**, arXiv:2405.04491.
144. Dauner, D.; Hallgarten, M.; Li, T.; Weng, X.; Huang, Z.; Yang, Z.; Li, H.; Gilitschenski, I.; Ivanovic, B.; Pavone, M.; et al. NAVSIM: Data-Driven Non-Reactive Autonomous Vehicle Simulation and Benchmarking. *arXiv* **2024**, arXiv:2406.15349.
145. Jia, X.; Yang, Z.; Li, Q.; Zhang, Z.; Yan, J. Bench2Drive: Towards Multi-Ability Benchmarking of Closed-Loop End-To-End Autonomous Driving. *arXiv* **2024**, arXiv:2406.03877.
146. Kazemkhani, S.; Pandya, A.; Cornelisse, D.; Shacklett, B.; Vinitzky, E. GPUDrive: Data-driven, multi-agent driving simulation at 1 million FPS. *arXiv* **2024**, arXiv:2408.01584.
147. Coscoy, M.; Zhou, Z.; Zhao, S.Z.; Wei, H.; Magtoto, A.; Liu, J.; Song, R.; Zimmer, W.; Huang, Z.; Tang, C.; et al. MDrive: Benchmarking Closed-Loop Cooperative Driving for End-to-End Multi-agent Systems. *arXiv* **2026**, arXiv:2605.10904.
148. Jayawardana, V.; Freydt, B.; Qu, A.; Hickert, C.; Yan, Z.; Wu, C. IntersectionZoo: Eco-driving for Benchmarking Multi-Agent Contextual Reinforcement Learning. *arXiv* **2024**, arXiv:2410.15221.
149. Bokade, R.; Jin, X. PyTSC: A Unified Platform for Multi-Agent Reinforcement Learning in Traffic Signal Control. *Sensors* **2024**, *25*, 1302.
150. Chang, H.-C.; Huang, S.-Y.; Chen, Y.-C.; Wu, I.-C. VissimRL: A Multi-Agent Reinforcement Learning Framework for Traffic Signal Control Based on Vissim. *arXiv* **2026**, arXiv:2601.18284.
151. Eichelbeck, M.; Markgraf, H.; Althoff, M. CommonPower: A Framework for Safe Data-Driven Smart Grid Control. *IEEE Trans. Smart Grid* **2024**, *17*, 71–82.

152. Marchesini, E.; Donnot, B.; Crozier, C.; Dytham, I.; Merz, C.; Schewe, L.; Westerbeck, N.; Wu, C.; Marot, A.; Donti, P.L. RL2Grid: Benchmarking Reinforcement Learning in Power Grid Operations. *arXiv* **2025**, arXiv:2503.23101.
153. Bogucki, D.; Lepak, Ł.; Parashar, S.; Błachowski, B.; Wawrzynski, P. EnEnv 1.0: Energy Grid Environment for Multi-Agent Reinforcement Learning Benchmarking. In Proceedings of the Adaptive Agents and Multi-Agent Systems, Detroit, MI, USA, 19–23 May 2025; pp. 361–370.
154. Salazar-Peña, N.; Tabares, A.; González-Mancera, A. MARLEM: A Multi-Agent Reinforcement Learning Simulation Framework for Implicit Cooperation in Decentralized Local Energy Markets. *arXiv* **2026**, arXiv:2602.16063.
155. Wu, Z.; Korolija, I.; Tang, R. MuFlex: A Scalable, Physics-based Platform for Multi-Building Flexibility Analysis and Coordination. *arXiv* **2025**, arXiv:2508.13532.
156. Chung, J.W.; Liang, Z.; Mao, Y.; Chen, J.; Chowdhury, M.; Dvorkin, V. OpenG2G: A Simulation Platform for AI Datacenter-Grid Runtime Coordination. *arXiv* **2026**, arXiv:2605.05519.
157. Demirel, G.; Grafenhorst, S.; Ohm, J.; Schäfer, B.; Förderer, K.; Hagenmeyer, V. Multi-Objective Reinforcement Learning with Shielding for Control and Safety in Power Grids. In *ACM SIGENERGY Energy Informatics Review*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 93–104.
158. Beukman, M.; Ingram, B.; Tasse, G.N.; Rosman, B.; Ranchod, P. RobocupGym: A challenging continuous control benchmark in Robocup. *arXiv* **2024**, arXiv:2407.14516.
159. Li, Z.; Bjelonic, F.; Klemm, V.; Hutter, M. MARLadona—Towards Cooperative Team Play Using Multi-Agent Reinforcement Learning. In Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA), Atlanta, GA, USA, 19–23 May 2025; pp. 15014–15020.
160. Orsula, A.; Geist, M.; Olivares-Mendez, M.; Martínez, C. Space Robotics Bench: Robot Learning Beyond Earth. *arXiv* **2025**, arXiv:2509.23328.
161. Oliveira, A.; Dyreby, K.; Caldas, F.; Soares, C. OrbitZoo: Real Orbital Systems Challenges for Reinforcement Learning. *arXiv* **2025**, arXiv:2504.04160.
162. Schwartz, D.; Hansson, A.; Bodmer, S.; Sternberg, D.; Jia-Richards, O.; Albee, K. SmallSatSim: A High-Fidelity Simulation and Training Toolkit for Microgravity Robotic Close Proximity Operations. *arXiv* **2026**, arXiv:2603.14598.
163. Mahajan, A.; Rashid, T.; Samvelyan, M.; Whiteson, S. Maven: Multi-agent variational exploration. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 7613–7624.
164. Yang, Y.; Hao, J.; Liao, B.; Shao, K.; Chen, G.; Liu, W.; Tang, H. QAtten: A general framework for cooperative multiagent reinforcement learning. *arXiv* **2020**, arXiv:2002.03939.
165. Baker, B.; Kanitscheider, I.; Markov, T.; Wu, Y.; Powell, G.; McGrew, B.; Mordatch, I. Emergent Tool Use From Multi-Agent Autocurricula. *arXiv* **2020**, arXiv:1909.07528.
166. Wu, C.; Kreidieh, A.; Parvate, K.; Vinitzky, E.; Bayen, A.M. Flow: Architecture and benchmarking for reinforcement learning in traffic control. *arXiv* **2017**, arXiv:1710.05465.
167. Vinitzky, E.; Parvate, K.; Kreidieh, A.; Wu, C.; Bayen, A. Lagrangian control through deep-rl: Applications to bottleneck decongestion. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, Maui, HI, USA, 4–7 November 2018; pp. 759–765.
168. Wei, H.; Xu, N.; Zhang, H.; Zheng, G.; Zang, X.; Chen, C.; Zhang, W.; Zhu, Y.; Xu, K.; Li, Z. Colight: Learning network-level cooperation for traffic signal control. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1913–1922.
169. Nasir, Y.S.; Guo, D. Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 2239–2250.
170. Lazic, N.; Boutilier, C.; Lu, T.; Wong, E.; Roy, B.; Ryu, M.; Imwalle, G. Data center cooling using model-predictive control. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 3818–3827.
171. Nikolaidis, S.; Hsu, D.; Srinivasa, S. Human-robot mutual adaptation in collaborative tasks: Models and experiments. *Int. J. Robot. Res.* **2017**, *36*, 618–634.
172. Carroll, M.; Shah, R.; Ho, M.K.; Griffiths, T.; Seshia, S.; Abbeel, P.; Dragan, A. On the utility of learning about humans for human-ai coordination. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 5174–5185.
173. Peng, X.B.; Andrychowicz, M.; Zaremba, W.; Abbeel, P. Sim-to-real transfer of robotic control with dynamics randomization. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Brisbane, Australia, 21–25 May 2018; pp. 3803–3810.
174. Pinto, L.; Davidson, J.; Sukthankar, R.; Gupta, A. Robust adversarial reinforcement learning. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 2817–2826.
175. Al-Shedivat, M.; Bansal, T.; Burda, Y.; Sutskever, I.; Mordatch, I.; Abbeel, P. Continuous adaptation via meta-learning in nonstationary and competitive environments. *arXiv* **2017**, arXiv:1710.03641.

176. Tessera, K.; Hinckeldey, L.; Zamboni, R.; Abel, D.; Storkey, A. Remembering the Markov Property in Cooperative MARL. *arXiv* **2025**. arXiv:2507.18333.
177. Guo, J.; Chen, Y.; Hao, Y.; Yin, Z.; Yu, Y.; Li, S. Towards Comprehensive Testing on the Robustness of Cooperative Multi-agent Reinforcement Learning. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), New Orleans, LA, USA, 19–20 June 2022. <https://doi.org/10.1109/CVPRW56347.2022.00022>.
178. Jing, Z.; Chang, X.; Zhu, M.; Li, S.; Liu, A.; Li, X.; Liu, X. A Comprehensive Evaluation Framework for Multi-Agent Reinforcement Learning. In Proceedings of the DAI '24: Proceedings of the 2024 6th International Conference on Distributed Artificial Intelligences, Singapore, 18–22 December 2024. <https://doi.org/10.1145/3719545.3759089>.
179. Rashid, T.; Samvelyan, M.; de Witt, C.S.; Farquhar, G.; Foerster, J.; Whiteson, S. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *arXiv* **2018**, arXiv:1803.11485.
180. Hu, T.M. Multi-Agent Reinforcement Learning in Starcraft: Algorithmic Advances and Collaborative Intelligence Challenges. In Proceedings of the International Conference on Computer Science and Electronic Information Technology (CSEIT 2025), Toronto, ON, Canada, 19–20 July 2025. <https://doi.org/10.1051/itmconf/20257801013>.
181. Gu, S.; Yang, L.; Du, Y.; Chen, G.; Walter, F.; Wang, J.; Knoll, A. A review of safe reinforcement learning: Methods, theory and applications. *arXiv* **2021**, arXiv:2205.10330.
182. Taylor, M.E.; Stone, P. Transfer learning for reinforcement learning domains: A survey. *J. Mach. Learn. Res.* **2009**, *10*, 1633–1685.
183. Singh, S.; Mahjoub, O.; de Kock, R.; Khelifi, W.; Vall, A.; Tessera, K.; Pretorius, A. How much can change in a year? Revisiting Evaluation in Multi-Agent Reinforcement Learning. *arXiv* **2023**, arXiv:2312.08463.
184. Papadopoulos, G.; Kontogiannis, A.; Papadopoulou, F.; Poulianos, C.; Koumentis, I.; Vouros, G.A. An Extended Benchmarking of Multi-Agent Reinforcement Learning Algorithms in Complex Fully Cooperative Tasks. *arXiv* **2025**, arXiv:2502.04773.
185. Agarwal, R.; Schwarzer, M.; Castro, P.S.; Courville, A.; Bellemare, M.G. RLiable: Better evaluation for reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 11766–11779.
186. Foerster, J.; Nardelli, N.; Farquhar, G.; Afouras, T.; Torr, P.H.; Kohli, P.; Whiteson, S. Stabilising experience replay for deep multi-agent reinforcement learning. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 1146–1155.
187. Shoham, Y.; Leyton-Brown, K. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*; Cambridge University Press: Cambridge, MA, USA, 2008.
188. Wooldridge, M. *An Introduction to MultiAgent Systems*, 2nd ed.; Wiley: Hoboken, NJ, USA, 2009.
189. Stone, P.; Veloso, M. Multiagent systems: A survey from a machine learning perspective. *Auton. Robot.* **2000**, *8*, 345–383.
190. Panait, L.; Luke, S. Cooperative Multi-Agent Learning: The State of the Art. *Auton. Agents Multi-Agent Syst.* **2005**, *11*, 387–434.
191. Durfee, E.H., Distributed problem solving and planning. In *Mutli-Agents Systems and Applications*; Springer: Berlin/Heidelberg, Germany, 2001; p. 118–149.
192. van Otterlo, M.; Wiering, M. Reinforcement Learning and Markov Decision Processes. In *Reinforcement Learning. Adaptation, Learning, and Optimization*; Wiering, M., van Otterlo, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 12.
193. Matsuo, Y.; LeCun, Y.; Sahani, M.; Precup, D.; Silver, D.; Sugiyama, M.; Uchibe, E.; Morimoto, J. Deep learning, reinforcement learning, and world models. *Neural Netw.* **2022**, *152*, 267–275.
194. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
195. Mienye, I.; Swart, T. A Comprehensive Review of Deep Learning: Architectures, Recent Advances, and Applications. *Information* **2024**, *15*, 755.
196. Mikolov, T.; Karafiát, M.; Burget, L.; Černocký, J.; Khudanpur, S. Recurrent neural network based language model. In Proceedings of the Proc. Interspeech 2010, Chiba, Japan, 26–30 September 2010; pp. 1045–1048.
197. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828.
198. Cummins, C.; Petoumenos, P.; Wang, Z.; Leather, H. End-to-End Deep Learning of Optimization Heuristics. In Proceedings of the 26th International Conference on Parallel Architectures and Compilation Techniques (PACT), Portland, OR, USA, 9–13 September 2017; pp. 219–232.
199. Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; de Las Casas, D.; Hendricks, L.A.; Welbl, J.; Clark, A.; et al. Training Compute-Optimal Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 10–16 December 2023; Volume 35, pp. 30016–30030.
200. François-Lavet, V.; Henderson, P.; Islam, R.; Bellemare, M.G.; Pineau, J. An Introduction to Deep Reinforcement Learning. *Found. Trends Mach. Learn.* **2018**, *11*, 219–354.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.