Scaling Up Multi-Agent Reinforcement Learning for Large Agent Teams and Long-Horizon Tasks: A Survey

MINGHONG GENG, School of Computing and Information Systems, Singapore Management University, Singapore, Singapore
SHUBHAM PATERIA, School of Computing and Information Systems, Singapore Management University, Singapore, Singapore
BUDHITAMA SUBAGDJA, School of Computing and Information Systems, Singapore Management University, Singapore, Singapore
AH HWEE TAN, School of Computing and Information Systems, Singapore Management University, Singapore, Singapore

Multi-agent reinforcement learning (MARL) empowers multiple autonomous agents to acquire effective policies for collaborative problem-solving. Over the last decade, MARL has seen significant advancements, with numerous algorithms achieving impressive performance across various benchmarks and real-world applications. Nevertheless, the scalability of multi-agent systems, in terms of the number of agents and the length of the task horizon, remains a critical consideration for applying MARL methods to complex problem-solving. Given that a dedicated review of the existing approaches and challenges in scaling up multi-agent systems remains largely absent, this survey aims to bridge this gap by delivering a comprehensive review of MARL methods developed to tackle challenging, scaled-up tasks. To this end, a novel taxonomy of MARL studies is introduced, categorizing them based on the external organizational control structures over all agents and the internal policy structures of individual agents. The survey also discusses the scales of popular MARL environments and tasks, providing a snapshot of the current challenging problems of interest. Furthermore, this survey underscores a set of critical open problems that call for further investigation in the field of scalable MARL.

CCS Concepts: • **Computing methodologies** → **Multi-agent reinforcement learning**; **Multi-agent systems**; **Cooperation and coordination**.

Additional Key Words and Phrases: Multi-agent reinforcement learning, Scaling up MARL, Long-horizon

1 Introduction

Multi-agent reinforcement learning (MARL) has emerged as a potent learning paradigm for addressing complex problems necessitating collaborative decision-making among multiple agents. MARL agents learn policies through iterative interactions with the environment, aiming to maximize their cumulative future rewards. Various real-world problems are suited for multi-agent modeling, such as adaptive traffic signal control [17, 53, 118, 133], autonomous vehicle control [140], video games [98], network packet routing [109], and multi-robot trash collection [67]. However, the majority of current MARL studies deal with comparatively small-scale problems characterized by either a small number of agents or a short *time horizon* for decision-making. Here, the *time horizon* denotes the number of time steps over which agents strategize or execute a course of action to complete

Authors' address: Minghong Geng, mhgeng@smu.edu.sg; Shubham Pateria, shubhamp@smu.edu.sg; Budhitama Subagdja, budhitamas@smu.edu.sg; Ah-Hwee Tan, ahtan@smu.edu.sg, School of Computing and Information Systems, Singapore Management University, 80 Stamford Rd, Singapore, 178902.

ACM 1557-7341/2026/6-ART
https://doi.org/10.1145/3817113

Table 1. Comparative analysis of MARL surveys. A checkmark (✓) indicates comprehensive coverage of the topic, while a partial checkmark (✓*) indicates limited or partial coverage. Our survey uniquely focuses on scaling challenges from both team size and temporal horizon perspectives.

Survey	Scaling Challenges		Hierarchical Approaches		Learning Paradigms			Theoretical Analysis	Published Year
	Large Teams	Long Horizon	Control Hierarchy	Policy Hierarchy	DTDE	CTDE	CTCE		
Weiß [129]					✓				1995
Buşoniu et al. [12]					✓	✓*	✓*	✓	2010
Matignon et al. [74]					✓			✓	2012
Silva and Costa [105]	✓*				✓	✓	✓		2019
Hernandez-Leal et al. [41]	✓*	✓*			✓	✓	✓	✓*	2019
Zhang et al. [150]					✓	✓	✓	✓	2021
This survey	✓	✓	✓	✓	✓	✓	✓	✓*	2026

DTDE: Decentralized Training with Decentralized Execution CTDE: Centralized Training with Decentralized Execution
CTCE: Centralized Training with Centralized Execution ✓*: Partial or limited coverage

a task [88]. As the scope of MARL methodologies and applications continuously broadens, effectively scaling up multi-agent learning methods to handle complex and expansive problems involving larger numbers of agents or longer horizons becomes a pressing challenge [16, 63, 110].

It is important to note that the relationship between scaling up MARL systems and problem difficulty is not straightforward. The challenge of scaling to larger agent teams depends not merely on the raw number of agents but on factors such as the degree of interdependence between agents, the complexity of joint action and state spaces, communication requirements, and the diversity of agent capabilities. This explains why certain benchmarks with fewer agents present greater scaling challenges than others with more agents. Structural scaling difficulties typically arise when coordination complexity grows non-linearly with the agent population, while temporal scaling difficulties emerge from sparse rewards, long causal chains between actions and outcomes, and complex dependencies across time steps.

Our survey delves into the complex landscape of scaling up multi-agent systems (MASs), dissecting it into two broad research challenges, namely **large agent teams** and **long-horizon tasks**. Effectively addressing both challenges is essential to unlock the potential of MARL in complex real-world scenarios. Nevertheless, the challenges and primary research directions for scaling up MASs have yet to be clearly defined. Traditional challenges in MARL, such as the *non-stationarity* [42, 86], the *curse of dimensionality* [11, 42], and *credit assignment* [2, 56], become increasingly complex and pronounced as MARL methods scale. Furthermore, navigating the extensive and expanding realm of MARL research to pinpoint strategies for effectively scaling up MAS presents an ongoing challenge. This demands an in-depth review to systematically understand the challenges and explore potential solutions.

Early survey papers by Weiß [129], Buşoniu et al. [12], and Matignon et al. [74] reviewed cooperative games and general MARL algorithms up to 2012. These studies laid the groundwork for understanding fundamental concepts and methodologies in cooperative multi-agent systems. Recently, Silva and Costa [105] conducted an in-depth investigation into the application of transfer learning in the context of MARL. By examining the utilization of transfer learning techniques, this survey sheds light on strategies for enhancing the adaptability and knowledge transfer capabilities of multi-agent systems. In a broader scope, Hernandez-Leal et al. [41] presented a comprehensive review of general multi-agent RL, encompassing competitive, cooperative, and mixed

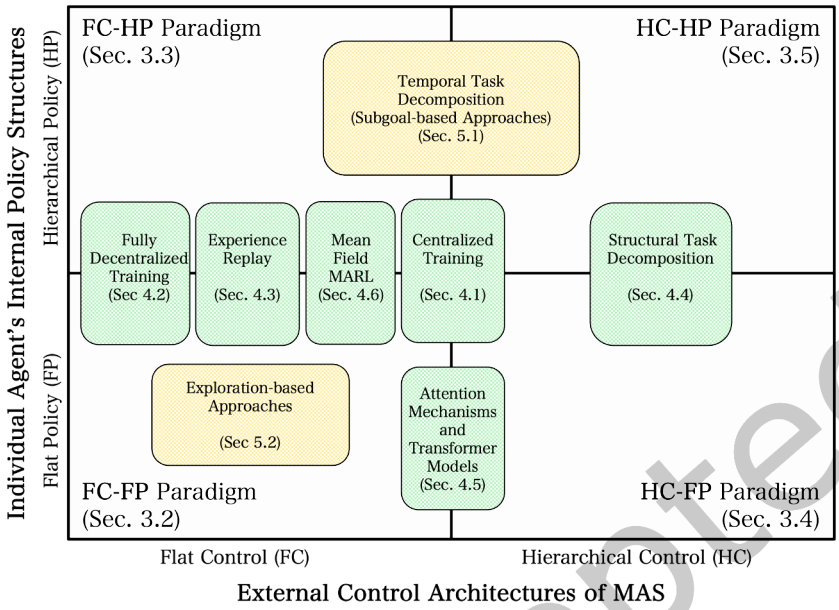

Fig. 1. The Multi-Agent Reinforcement Learning (MARL) taxonomy introduced in this paper. The reviewed MARL methods are classified based on the external control architectures of the entire multi-agent system (MAS), i.e., with flat or hierarchical control structures, and the individual agent's internal policy structures, i.e., with flat or hierarchical policy structures. The details of the four paradigms are presented in Section 3. The green boxes represent the methods for overcoming challenges related to increasing the number of agents in MAS, including the *non-stationarity* issue, the *curse of dimensionality*, and *structural credit assignment*, as presented in Section 4. The yellow boxes highlight methods for addressing challenges associated with long-horizon planning, particularly in terms of *temporal credit assignment*, as presented in Section 5.

environments. This survey also discussed practical challenges, including common implementation strategies, computational requirements, and unresolved questions. Zhang et al. [150] conducted an extensive review of theoretical results, convergence analyses, and complexity assessments of MARL algorithms, especially in the context of Markov games and extensive-form games across competitive, cooperative, and mixed environments. Notably, their work delved into the theoretical foundations of consensus and policy evaluation in cooperative scenarios. We summarize related preceding surveys in Table 1. While these surveys provide valuable insights into the understanding of MARL, our work distinctively focuses on the challenges and research directions related to scaling up MARL towards larger teams and longer time horizons.

We adopt a structured and systematic approach to present a literature review of the latest research progress in MARL, aiming to shed light on the diverse challenges, solutions, and future research directions towards scaling up MAS. We initiate our investigation by introducing a new taxonomy that categorizes MARL methods based on the **external control architectures of agents** and **internal policy structures learned by individual agents**, as illustrated in Figure 1. While a commonly used taxonomy classifies MARL methods based on their training and execution paradigms [34], it implicitly assumes that the multi-agent systems in question have neither complex hierarchical control architectures nor multi-level decision-making processes—an assumption that is overly simplistic and inapplicable to many recent MARL methods with hierarchical learning approaches. In this work, we classify MARL methods based on the control architectures and policy structures to cover a broader

scope of methods that are more relevant to scaling up MAS towards large teams and long horizons. Specifically, we organize MARL methods into four categories: *classic MARL*, *MARL with hierarchical control architectures over agents*, *MARL with agents learning hierarchical policies*, and *MARL with both hierarchical control architectures and policy structures*. This literature review demonstrates that MARL methods employing hierarchical control architectures play a pivotal role in the scalability of MARL methods for larger teams of agents. Concurrently, MARL approaches that involve agents learning hierarchical policies are closely linked with enhancing the scalability of MARL for longer time horizons.

For this survey, we reviewed approximately 120 papers on scaling up MARL published between 2015 and 2024. We identified relevant work through keyword searches in major academic databases using terms including "multi-agent reinforcement learning," "scaling," "large teams," "long horizon," and "hierarchical MARL." We also traced citation networks from seminal papers and previous surveys. Our selection focused on publications from top-tier AI venues, excluding papers that did not specifically address MARL scaling challenges or lacked practical scaling applications.

The rest of this paper is organized as follows. Section 2 presents the preliminaries of MARL. Section 3 introduces our proposed MARL taxonomy, while Sections 4 and 5 discuss the challenges and solutions for scaling to large agent teams and long-horizon tasks, respectively. Section 6 reviews popular MARL environments and benchmarks. Section 7 outlines open challenges and future directions. Finally, Section 8 concludes the survey by summarizing the key findings and highlighting important concerns in scaling up MARL.

2 MARL: Preliminaries

2.1 Markov Decision Process (MDP)

To present a clear road map of MARL, we first introduce foundational concepts of single-agent reinforcement learning relevant to understanding their multi-agent counterparts. Classic single-agent reinforcement learning is commonly modeled as a Markov Decision Process (MDP). Let $\mathcal{M} = (S, A, P, R, p_0, \gamma)$ be an MDP, where S is a set of states, A is a set of actions, P is a state-transition probability function where $P_a(s, s') = P(s_{t+1} = s' | s_t = s, a_t = a)$, that is, the probability of transiting from state s_t to s_{t+1} after agent taking action a at time step t, R is a reward function where $R_a(s, s')$ is the reward of taking action a that transits state s to s', p_0 is an initial state distribution, $\gamma \in [0, 1)$ is a reward discount factor. The goal of reinforcement learning is to find an action policy $\pi(a|s)$ that maximizes the expected discounted cumulative reward $E_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_{t+1}) | s_0 \sim p_0, a_t \sim \pi(a_t|s_t), s_{t+1} \sim P(s_{t+1}|s_t, a_t)\right]$, where s_0 is the initial state.

2.2 Partially Observable MDP (POMDP) and Decentralized POMDP (Dec-POMDP)

A classic assumption in the MARL literature is that agents have only partial sensing abilities and cannot observe the global state [22, 34, 150]. This means that agents may partially perceive different parts of the environment. Although sensing global states could assist agents in better decision-making, capturing global states in many practical scenarios might be challenging or infeasible. Therefore, a more realistic assumption is that agents make decisions based on local observations and operate under a *Partially Observable MDP* (POMDP) [59]. We formally define a POMDP as a tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \mathcal{O}, \gamma \rangle$, where $\mathcal{S}$ is set of global states, $\mathcal{A}$ is the set of actions, P is the state-transition probability, R is the reward function, $\mathcal{O}$ is the set of local/partial observations, and $\gamma \in [0, 1)$ is the reward discount factor.

Agents may operate in a decentralized cooperative multi-agent POMDP setting, taking actions solely based on their local observations. As a generalization of MDP and POMDP, Dec-POMDP [8, 93] models the coordination and decision-making process among multiple agents. Similar to the above definition of POMDP, we define Dec-POMDP as a tuple $\langle \mathcal{I}, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, \{\mathcal{R}_i\}_{i=1}^N, P, \{\mathcal{O}_i\}_{i=1}^N, \gamma \rangle$, where $\mathcal{I}$ represents the domain of N agents, $\mathcal{S}$ is the set of global states, P is the state-transition probability, $\{\mathcal{A}_i\}_{i=1}^N$, $\{\mathcal{R}_i\}_{i=1}^N$, and $\{\mathcal{O}_i\}_{i=1}^N$ are the set of actions,

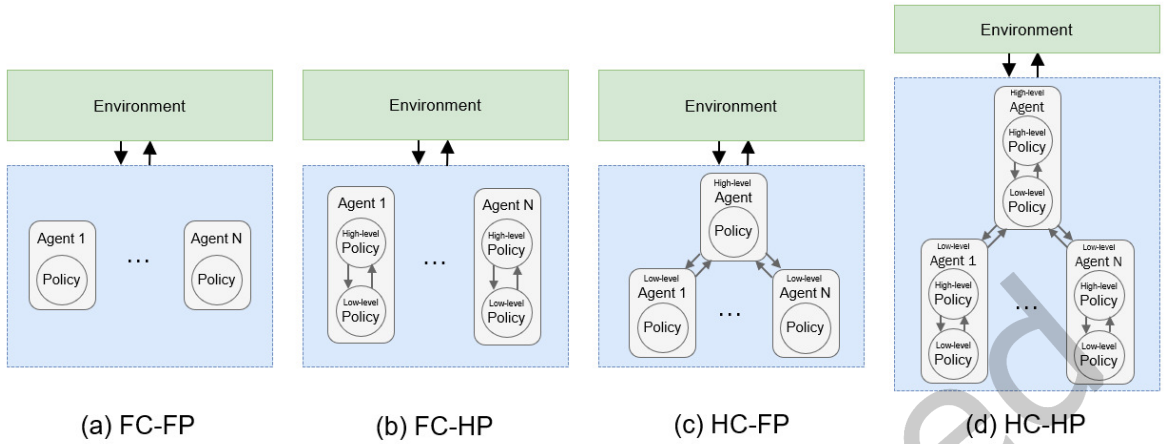

Fig. 2. An overview of four paradigms in multi-agent systems based on control architecture and policy structure. (a) Flat Control, Flat Policies (FC-FP): Agents operate without hierarchical control structures and employ non-hierarchical policies. (b) Flat Control, Hierarchical Policies (FC-HP): Agents lack hierarchical control structures but utilize hierarchical policies. (c) Hierarchical Control, Flat Policies (HC-FP): Agents adhere to a hierarchical managerial architecture where high-level controllers coordinate low-level agents. Each agent maintains an individual action policy. (d) Hierarchical Control, Hierarchical Policies (HC-HP): Agents follow a hierarchical managerial architecture and employ hierarchical policies for decision-making. This classification focuses on the external managerial architecture of multi-agent teams and the internal policy structures of individual agents. Other aspects, such as information-sharing mechanisms, are discussed in subsequent sections.

reward functions, and local/partial observations for all agents, respectively. At each time step in a Dec-POMDP, agent $i \in \mathcal{I}$ is only able to access a local observation $o_i \in \mathcal{O}_i$ and take action a_i sampled from policy $\pi_i(a_i|o_i)$, where π_i is the action policy of agent i. Agents jointly execute action $\boldsymbol{a} = \langle a_1, a_2, \ldots, a_N \rangle$, where $a_i \in \mathcal{A}_i$ and $i \in [1, N]$. A Dec-POMDP transits from state s to s' according to the transition function $P(s'|s, \boldsymbol{a})$. In a fully cooperative setting, all agents share the same global reward value $\mathcal{R}(s, \boldsymbol{a})$.

3 Control Architectures and Policy Structures in MARL

This section presents a comprehensive review of the landscape of MARL, analyzing its capabilities for scaling up multi-agent systems (MAS). To systematically organize the discussion, we introduce a novel taxonomy with two pivotal dimensions: (1) the control architecture organizing the agents and (2) the policy structures of individual agents.

For control architectures, we distinguish between hierarchical and non-hierarchical (flat) approaches. In hierarchical control, agents are organized into multiple levels, with higher-level agents coordinating lower-level agents, while in non-hierarchical control, all agents operate at the same level. For policy structures, hierarchical policies enable agents to make decisions at multiple temporal scales, with higher levels handling more abstract decisions and lower levels addressing immediate actions, enhancing agents' ability to address complex tasks with temporal abstraction [116].

3.1 Background: MARL Training and Execution Paradigms

The landscape of MARL has expanded considerably in recent years, offering various training and execution paradigms. Traditional MARL research classifies approaches based on training paradigms (*centralized* or *decentralized*) and execution paradigms (*centralized* or *decentralized*). Previous surveys [22, 34, 41, 81] have categorized

MARL into three paradigms: Decentralized Training with Decentralized Execution (DTDE), Centralized Training with Decentralized Execution (CTDE), and Centralized Training with Centralized Execution (CTCE).

This conventional taxonomy, however, has two significant limitations for scalability research: (1) it assumes all agents operate at the same decision-making level, neglecting emerging research on hierarchical MAS that can alleviate the curse of dimensionality; and (2) it overlooks hierarchical policy structures and temporal abstraction approaches that address sparse reward challenges. In light of these limitations of the conventional taxonomy, we thus propose a novel taxonomy based on control and policy structures as illustrated in Figure 2:

(1) **Flat Control, Flat Policies (FC-FP)**: The predominant learning paradigm in MARL research.
(2) **Flat Control, Hierarchical Policies (FC-HP)**: Extended from single-agent RL, primarily containing MAHRL.
(3) **Hierarchical Control, Flat Policies (HC-FP)**: Focused on hierarchical control architectures.
(4) **Hierarchical Control, Hierarchical Policies (HC-HP)**: The intersection of MAHRL and hierarchical MARL.

3.2 Flat Control, Flat Policies (FC-FP)

The FC-FP paradigm (Figure 2a) has emerged as a predominant focus in the MARL domain. Here, agents operate without hierarchical control structures, functioning at the same level and employing single-level policies. FC-FP methods widely adopt three schemes based on training and execution methodologies [34]: DTDE, CTCE, and CTDE. Table 2 summarizes representative FC-FP MARL methods discussed in this work.

The DTDE scheme represents a fundamental yet effective approach for scaling MAS. In the DTDE scheme, agents operate in a fully decentralized manner without communication, treating others as environmental components. Compared to centralized approaches, DTDE methods face fewer scalability constraints as agent numbers increase [84]. Notable examples include the work of Tampuu et al. [114], which extends DQN [77] to decentralized multi-agent settings, and IA2C [87] and IPPO [100], which implement independent Advantage Actor-Critic (A2C) [75] and Proximal Policy Optimization (PPO) [102] agents, respectively.

However, DTDE methods struggle in non-stationary environments, especially under partial observability. Centralized training approaches alleviate this by allowing agents to exchange information to enrich their local observations. In fully centralized schemes (CTCE), the MARL problem reduces to single-agent RL, with joint observations and actions aggregated as the state and action of a central agent [111]. CommNet [109] introduces a central controller that takes the state observations and the communication messages of all agents and performs multi-step communications to generate actions for all agents as its output. BiCNet [92] employs a vectorized actor-critic algorithm using shared actor and critic networks with a bi-directional recurrent neural network.

Notably, CTCE methods face the *curse of dimensionality* [11, 41] as combinatorial state-action spaces grow exponentially with the number of agents [34]. They may also encounter the "lazy agent" issue — a credit assignment problem where only some agents contribute while others remain inactive [54, 111]. CTDE approaches attempt to balance the benefits of both paradigms. Value decomposition methods such as VDN [111] decompose team rewards into individual rewards. QMIX [96] extends this with non-linear decomposition, while QTRAN [107] eliminates structural constraints to address a wider range of tasks. QPLEX [123] introduces a duplex dueling network architecture for efficient value function learning. Gupta et al. [37] extend three single-agent algorithms, namely DQN, DDPG [58], and TRPO [101], with parameter sharing. For sparse reward environments, methods such as Counterfactual Multi-Agent Policy Gradients (COMA) [30] and Lazy Agents Avoidance through Influencing External States (LAIES) [61] introduce specialized mechanisms for credit assignment. MAMBA [24] takes a different approach, leveraging model-based learning to improve sample efficiency.

Overall, FC-FP methods may struggle to scale to large agent numbers or long-horizon tasks, due to issues such as the curse of dimensionality, non-stationarity, and temporal credit assignment.

Table 2. Representative FC-FP methods discussed in this work.

Training and Execution Scheme	Method	Learning Algorithm for Individual Agents
DTDE	IQL [115]	Tabular Q-learning [127]
	IQL with DQN [114]	DQN [77]
	IA2C [87]	A2C [75]
	IPPO [100]	PPO [102]
CTCE	CommNet [109]	Not applicable since a single centralized algorithm is used
	BiCNet [92]	Not applicable since a single centralized algorithm is used
CTDE	VDN [111]	DQN [77]
	PS-TRPO [37]	TRPO [101]
	PS-DQN [37]	DQN [77]
	PS-DDPG [37]	DDPG [58]

Table 3. An overview of Flat Control, Hierarchical Policy (FC-HP) models and algorithms.

Training Scheme	Execution Scheme	Model/Algorithm	Agents' Action Policy	
			High-level	Low-level
Decentralized	Decentralized	h-IL [116]	DQN	DQN
Centralized	Decentralized	h-Comm [116]	DQN	DQN
		h-Qmix [116]	DQN	DQN
		ROMA [125]	Role Encoder and Role Decoder	Local Utility Network
		RODE [126]	Feedforward Network (Role Selector)	Feedforward Network (Role Policy)
		HAVEN [138]	Value Mixing Network and Macro Mixing Network	Agent Network

3.3 Flat Control, Hierarchical Policies (FC-HP)

The FC-HP paradigm (Figure 2b) integrates hierarchical reinforcement learning with MARL, enabling agents to learn hierarchical policies rather than single-level ones. This paradigm (also known as MAHRL [90]) maintains agents at the same organizational level but allows them to decompose tasks temporally.

Multi-agent MAXQ [67] pioneers this approach by using MAXQ [21] hierarchies to decompose tasks into subtasks. More recently, Tang et al. introduce three MAHRL algorithms [116]: h-IL (hierarchical independent learning), h-Comm (hierarchical communication), and h-QMIX. Each extends a corresponding flat approach [96, 109, 114] with temporal abstraction capabilities. h-IL implements fully decentralized learning with two-level hierarchical policies, where high-level policies select temporal abstractions and low-level policies choose primitive actions. h-Comm and h-QMIX incorporate centralized learning with inter-agent communication at higher policy levels. The Hierarchical Skill Discovery (HSD) [141] algorithm employs a shared centralized high-level policy with independent low-level policies. HAVEN [138] introduces a hierarchical off-policy value decomposition framework inspired by human nervous system coordination mechanisms. Role-oriented approaches such as

ROMA [125] and RODE [126] introduce dynamic specialized roles to enable similar agents to share learning experiences, with roles represented as stochastic latent variables. RODE extends this by efficiently discovering roles through clustering actions based on their environmental effects. Overall, FC-HP methods enable scaling to longer-horizon problems by decomposing tasks into more manageable subtasks, simplifying exploration and learning. The representative FC-HP methods are presented in Table 3.

3.4 Hierarchical Control, Flat Policies (HC-FP)

The HC-FP paradigm (Figure 2c) introduces hierarchical control architectures while maintaining flat policies for individual agents. Inspired by social structures such as feudal hierarchies [3], master-slave architectures [50], coach-player paradigms [141], and human nervous systems [138], these approaches organize agents into control hierarchies.

HC-FP systems typically have agents at different hierarchy levels specializing in tasks with varying temporal and spatial complexities. Higher-level agents observe larger state spaces and provide temporally extended goals or instructions to lower-level agents who execute primitive actions. This reduces observation-action space complexity and mitigates the curse of dimensionality. Consequently, agents at different hierarchical levels operate with distinct observation and action spaces. Agents at higher levels are commonly designed with the role of managers [3], meta-controllers [52], master controllers [50], or conductors [103]. Their actions consist of temporally extended goals or instructions that must be accomplished by the lower-level agents, and they commonly do not directly interact with the environment [3, 50, 52, 103]. In contrast, the lower-level agents take the roles of executors, receiving instructions from the high-level agents and executing primitive actions in the environment.

Feudal Multiagent Hierarchies (FMH) [3] implements a two-level hierarchy with manager agents setting subgoals for worker agents. FMH employs pre-training with the manager selecting from predefined subgoals while worker agents (implemented as DDPG [58]) gain experience. This bootstrapped learning approach accelerates coordination. Federated Control with Reinforcement Learning (FCRL) [52] organizes DQN [76] agents into a meta-controller and multiple controllers operating at different temporal scales. The meta-controller allocates subtasks to controller pairs by defining constraints, while controllers execute actions to satisfy these constraints. Parameter sharing among controllers improves scalability, although limitations appear beyond six controllers [81]. Other approaches include ALMA [46] with allocation and execution controllers, HAMMER [38] with high-level messaging to lower-level agents, and MS-MARL [50] with RNN-based master-slave modules that share hidden states. VAST [94] addresses the performance bottleneck in value function factorization by approximating factorization for agent subgroups.

Overall, HC-FP methods facilitate scaling to both larger agent teams and longer horizons through their rich hierarchical structures that enable temporal abstraction, role decomposition, and manager-worker control.

3.5 Hierarchical Control, Hierarchical Policies (HC-HP)

The Hierarchical Control, Hierarchical Policies (HC-HP) paradigm, as depicted in Figure 2d, employs a hierarchical managerial structure that coordinates agents, each of which has hierarchical policies. This paradigm leverages the advantages of both external and internal hierarchical control within multi-agent systems, making it an ideal framework for expanding the scope of MARL to encompass larger agent teams and to tackle complex, long-horizon tasks. Despite its potential, the current MARL literature predominantly focuses on the FC-FP, FC-HP, and HC-FP paradigms, with fully hierarchical MARL, as represented by the HC-HP paradigm, remaining relatively rare. Practically, HC-HP algorithms can integrate techniques from other paradigms to substantially enhance their capabilities. In this study, we highlight the significance of the HC-HP learning paradigm and encourage further exploration of this approach to facilitate the scalability of MARL.

4 Towards Large Agent Teams: Challenges and Approaches

The number of autonomous agents is a fundamental concern in designing MAS. Many real-world scenarios require large numbers of agents. For instance, applications such as traffic signal control [14, 17, 133, 149] and multi-vehicle systems control [140], particularly in urban settings, often require the simultaneous coordination of hundreds or thousands of learning agents. Moreover, the capacity of a MAS is intrinsically associated with the number of its participating agents. Large-scale MAS generally yields greater advantages than smaller ones [115]. These advantages are particularly pronounced when agents must engage with adversaries, such as in multi-agent combat simulations [25, 98]. In this work, we categorize the studies on increasing the number of agents as *structurally scaling up MAS*, in contrast to the work on *temporally scaling up MAS*, discussed in Section 5.

Structurally scaling up MAS poses several significant challenges to be addressed, primarily including the *non-stationary* nature [41, 86] of the multi-agent environments and the complexities associated with multi-agent exploration across a vast joint state-action space [110, 142], where the latter issue is also known as the *curse of dimensionality* [11, 41], caused by the *combinatorial nature* [42, 150] of MARL. In practice, the non-stationarity and the curse of dimensionality represent two crucial, interlinked challenges of structurally scaling up MAS. The non-stationarity stems from the simultaneous policy optimization by agents in partially observable environments, affecting the predictability and consistency of the environment's dynamics. To address this, *centralized training* (see Section 3) has emerged as a prominent strategy, demonstrating success in various implementations [96]. However, this approach exacerbates the curse of dimensionality, as the joint state and action space expands exponentially with the number of agents in centralized training frameworks. Consequently, finding a balance between these issues necessitates a well-calibrated approach to cooperation and communication among agents. This dilemma continues to be an active area of exploration in the MARL community.

While discussing the prevalent challenges of MARL is a common theme in surveys, such as those by Hernandez-Leal et al. [42], Nguyen et al. [81], Zhang et al. [150], and Gronauer and Diepold [34], a focused review on structurally scaling up MAS is notably lacking. Existing studies occasionally use the term *scalability issue* [150] to broadly refer to challenges in structurally scaling up MAS. However, this usage is somewhat imprecise, as the scalability in MARL encompasses not only structural aspects but also temporal scaling, another significant area of MARL studies (see Section 5). Moreover, structural scaling involves a multitude of complex challenges, each constituting its own intricate domain of study. This section delves into the contemporary strategies devised to tackle two critical challenges, i.e., the non-stationarity and the curse of dimensionality, arising from structurally scaling up MAS. We recommend readers refer to the aforementioned surveys for broader insights into MARL challenges beyond structural scaling, especially the surveys by Hernandez-Leal et al. [41] and Papoudakis et al. [86] for a specific discussion on the non-stationarity problem of MARL.

4.1 Centralized Training Methods

Centralized training is one of the most widely adopted approaches to address the non-stationarity issue when learning multiple policies simultaneously. In this approach, agents are allowed to share or receive information, for example, the joint observation and joint action of all other agents. As a result, each agent's policy is conditioned on the observations, actions, or rewards of others; thereby, the non-stationarity of multi-agent environments can be reduced. Based on the execution scheme (how agents operate after training), centralized MARL methods can be divided into two classes, namely *centralized training with centralized execution* (CTCE) [34], and *centralized training with decentralized execution* (CTDE), where the former is also known as the *fully centralized* scheme. In the CTCE scheme, multi-agent learning problems could be essentially reduced to single-agent reinforcement learning problems, as the joint observation and joint action of all agents are aggregated as the state and action of a central agent [111]. Therefore, single-agent RL methods, such as the actor-critic methods [75] or policy gradient methods [102], could be applied [34]. While the CTCE scheme reduces the non-stationarity, it suffers from the

Table 4. The control architectures, policy structures, and the training and execution paradigms of centralized MARL methods.

MARL Method	Control Architecture and Policy Structure	High-level Policy (if any)		Low-level Policy	
		Training Paradigm	Execution Paradigm	Training Paradigm	Execution Paradigm
CommNet [109]	Flat Control, Flat Policies (FC-FP)	N.A.	N.A.	Centralized	Centralized
BiCNet [92]	Flat Control, Flat Policies (FC-FP)	N.A.	N.A.	Centralized	Centralized
COMA [30]	Flat Control, Flat Policies (FC-FP)	N.A.	N.A.	Centralized	Decentralized
MAA2C [65]	Flat Control, Flat Policies (FC-FP)	N.A.	N.A.	Centralized	Decentralized
MADDPG [65]	Flat Control, Flat Policies (FC-FP)	N.A.	N.A.	Centralized	Decentralized
RIAL [29]	Flat Control, Flat Policies (FC-FP)	N.A.	N.A.	Centralized	Decentralized
DIAL [29]	Flat Control, Flat Policies (FC-FP)	N.A.	N.A.	Centralized	Decentralized
VDN [111]	Flat Control, Flat Policies (FC-FP)	N.A.	N.A.	Centralized	Decentralized
QMIX [96]	Flat Control, Flat Policies (FC-FP)	N.A.	N.A.	Centralized	Decentralized
h-Comm [116]	Flat Control, Hierarchical Policies (FC-HP)	Centralized	Centralized	Decentralized	Decentralized
h-QMIX [116]	Flat Control, Hierarchical Policies (FC-HP)	Centralized	Decentralized	Decentralized	Decentralized
HSD [141]	Flat Control, Hierarchical Policies (FC-HP)	Centralized	Decentralized	Decentralized	Decentralized
HAMMER [38]	Flat Control, Hierarchical Policies (FC-HP)	Centralized	Decentralized	Centralized	Decentralized
MS-MARL [50]	Hierarchical Control, Flat Policies (HC-FP)	Independent	Independent	Decentralized	Decentralized
ALMA [46]	Hierarchical Control, Flat Policies (HC-FP)	Independent	Independent	Centralized	Decentrazlied

curse of dimensionality [11, 41], as the joint state-action space and the interactions between agents increase exponentially by the number of agents, making learning and sufficient exploration a challenging problem [34]. The CTDE scheme endeavors to strike a balance between fully centralized and fully decentralized schemes. The core idea is to use decentralized agents instead of a centralized one, which can train using shared information but operate in a decentralized manner after training. Consequently, each CTDE agent deals with a smaller state-action space in contrast to the large combinatorial space in the case of a centralized CTCE agent. Various existing surveys have reviewed MARL studies under the CTCE and CTDE schemes, such as the work by Gronauer et al. [34], Du and Ding [22], and Wong et al. [131]. Unlike these surveys, this subsection reviews important centralized MARL studies in terms of control and policy structures and the structural scale of MAS. We hereby encourage interested readers to refer to the above surveys for more insights into MARL's training and execution schemes.

Table 4 presents the representative centralized MARL methods published in recent years. Notably, most centralized MARL methods follow the FC-FP paradigm, assuming agents cooperate on the same level of hierarchy and learn single-level policies. For example, fully centralized methods, including CommNet [109] and BiCNet [92] aim at enhancing multi-agent communications and aggregate the information from all individual agents for a central controller, which subsequently selects actions for each agent; CTDE methods, such as COMA [30], MAA2C [65], MADDPG [65], RIAL [29], DIAL [29], VDN [111], QMIX [96], etc., focus on different aspects of MARL and many of them achieve state-of-the-art performance on various MARL benchmarks. As these FC-FP methods were discussed in Section 3.2, here we will not introduce these methods in detail.

Conventional discussions of CTCE, CTDE, and DTDE typically fall within the FC-FP framework, where agent relationships are non-hierarchical. However, this conventional framework fails to capture the complexity of hierarchical methods, where internal and external interactions among agents and their policies introduce additional layers of complexity. This complexity is particularly evident in paradigms like the FC-HP, where agents are characterized by hierarchical policies. In such scenarios, each policy level can have individually tailored training and execution strategies. Intuitively, the greater the number of policy levels an agent possesses, the wider the range of possible centralized and decentralized combinations available for consideration in MARL methodologies. As a result, the potential number of training and execution schemes increases with the number of policy levels an agent has. For instance, consider a MAS where agents have bi-level policies. Since each level can independently adopt one of three schemes—CTCE, CTDE, or DTDE—this results in nine distinct training-execution combinations, ranging from the fully centralized (CTCE-CTCE) to the fully decentralized (DTDE-DTDE). Similar complexities are also observable in hierarchical multi-agent learning methods within the

Table 5. A selective overview of experience replay methods discussed in this work.

MARL Method	Control Architecture and Policy Structure	Descriptions
Foerster et al. [28]	Flat Control, Flat Policies (FC-FP)	Employ a multi-agent variant of *importance sampling* to decay outdated data; conditions each agent's value function on a *fingerprint* that disambiguates the age of data.
Palmer et al. [85]	Flat Control, Flat Policies (FC-FP)	Augment experience samples with *leniency* to restraint the update of state-action values.
SRS [153]	Flat Control, Flat Policies (FC-FP)	Use a *scheduled replay strategy* (SRS) to determine sampling weights based on an experience's position within a trajectory.
ACER [116]	Flat Control, Hierarchical Policies (FC-HP)	Augment high-level transitions with sub-transitions and sampling piles of concurrent experiences for all agents.
MASER [48]	Flat Control, Hierarchical Policies (FC-HP)	Formulate subgoals using the multi-agent experience replay buffer and considers both individual and joint Q-values of agents

HC-FP paradigm. While most hierarchical methods currently focus on bi-level architectures, expanding these methods will inevitably involve navigating the complexities associated with multiple hierarchical levels.

In the FC-HP paradigm, agents do not have hierarchical control architectures, but each agent employs multi-level policies. For example, agents in h-Comm [116], h-QMIX [116], HSD [141], and HAMMER [38] methods commonly have bi-level policies. In h-Comm, the training and execution of the high-level policies are fully centralized, while in h-QMIX, HSD, and HAMMER, the training of high-level policies is in a centralized manner, but their execution is decentralized. Notably, the low-level policies of agents in the FC-HP paradigm are commonly fully decentralized, except for HAMMER. The majority of methods in the HC-FP paradigm focus on two levels of control, with one higher-level controller and multiple low-level controllers. Consequently, the policy of the high-level controller is commonly trained independently, but the high-level controller may receive information from low-level agents.

4.2 Fully Decentralized Training Methods

As introduced above, in centralized MARL methods, information sharing may happen in various places. Unlike centralized methods, decentralized MARL methods do not have information exchange at any level of the control hierarchy or the agent's internal policy. Similar to centralized MARL methods, the majority of decentralized training methods lie in the category of FC-FP, for example, the DTDE methods like IQL [114, 115], IA2C [87], IPPO [100]. Under the fully decentralized FC-HP category, h-IL [116] is a key method that enables agents to have multi-level policies and operate in a fully decentralized manner. MS-MARL [50] is an HC-FP method but has fully decentralized operations on the lower-level policies. However, as the high-level controller in MS-MARL aggregates the state and observation of all lower-level controllers for training and execution, we hereby classify HC-FP methods as predominantly centralized methods.

4.3 Experience Replay

Experience replay is a technique extensively utilized in off-policy reinforcement learning, wherein agents' *experiences* are stored within a *replay buffer* and subsequently sampled as mini-batches to train reinforcement learning models. Due to the non-stationarity of multi-agent environments where agents update policies concurrently, the experiences captured in the buffer are continually generated by rapidly-adapting policies. In this case, the experiences sampled may no longer reflect the current policy and may be detrimental to multi-agent learning [28, 153].

Earlier endeavors on multi-agent experience replay predominantly align with the FC-FP paradigm. Among the pioneering efforts, Foerster et al. [28] propose two methods to address the challenge of multi-agent experience replay. The first method employs a multi-agent variant of *importance sampling* to downweight outdated data, while the second method conditions each agent's value function on a *fingerprint* that disambiguates the age of data from the replay buffer. Palmer et al. augment samples with a *leniency* metric, which diminishes in response to the frequency of state-action pair visits [85]. The state-action values are updated only if the leniency metric falls below a randomly drawn variable $x \sim U(0, 1)$. As a result, the frequency of updates to state-action pairs is indirectly regulated by experiences stored in the replay buffer. Prioritized experience replay (PER) [99] is an experience replay technique that adjusts sampling probabilities to facilitate multi-agent policy training. However, vanilla PER struggles in noisy multi-agent environments as the rewards might not accurately reflect the true value of specific actions. Zhang et al. propose a *scheduled replay strategy* (SRS) [153] to improve PER by utilizing pre-determined *rising schedules* to determine varying priorities based on an experience's position within a trajectory. Through SRS, experiences nearer to terminal states are prioritized, aiming to significantly reduce the estimation bias for samples close to these terminal states.

Some recent work considers multi-agent experience replay methods in the context of hierarchical learning. Compared with MARL methods learning non-hierarchical policies, methods learning hierarchical policies encounter more challenges. During the training of hierarchical policies, the experience samples may be captured disproportionately across all levels of the hierarchy, particularly when higher-level policies tend to operate less frequently than their lower-level counterparts, resulting in sparse replay buffers for the high-level policies. To address the above issue, Tang et al. propose an *augmented concurrent experience replay* (ACER) strategy [116] under the FC-HP paradigm. ACER enhances learning by augmenting high-level transitions with sub-transitions and sampling batches of concurrent experiences for all agents. ACER allows for more efficient updates from denser experiences and encourages coordinated policy learning among agents, thereby stabilizing the training process. Multi-agent experience replay methods can also facilitate hierarchical policy learning by generating subgoals. Jeon et al. propose a method named *MARL with subgoals generated from experience replay buffer* (MASER) [48]. MASER formulates subgoals using the multi-agent experience replay buffer and considers both individual and joint Q-values of agents. MASER considers the intrinsic rewards of each agent and trains agents' policies following the CTDE paradigm. By autonomously generating subgoals from the experience replay buffer, MASER balances the objectives of individual agents and the joint Q-value. Table 5 summarizes the multi-agent experience replay methods reviewed in this subsection.

4.4 Structural Task Decomposition

As *the curse of dimensionality* is exacerbated by the increasing number of agents, a natural approach is to group agents into multiple independent subgroups, allowing certain communication mechanisms, thereby reducing the exploration space of each subgroup and improving the learning efficiency. This category of methods essentially performs a team-wise task decomposition by reducing the large-scale multi-agent tasks into a coordination problem of multiple smaller subgroups of RL agents. Therefore, we name this approach *structural task decomposition*, differentiating it from the *temporal task decomposition* methods introduced in Section 5.1. As introduced in Section 5.1, some structural task decomposition methods are combined with temporal task decomposition methods for scaling up MARL. In this subsection, we especially present the work that only focuses on grouping agents into subgroups. A comprehensive overview of structural task decomposition methods can be found in Table 6.

Earlier methods in structural task decomposition for MARL primarily employed rule-based decomposition, where subgroup formation was manually determined. A basic approach within this paradigm involves treating each agent as an individual subgroup, often coordinated by an additional controller agent. Such methods fall

under the *Hierarchical Control, Flat Policies* (HC-FP) paradigm, detailed in Section 3.4. For example, the Feudal Multi-agent Hierarchies (FMH) framework [3] organizes two agents into separate subgroups using heuristics. The FCRL method [52] extends this approach to systems with two to six agents, incorporating inter-agent communication. Other rule-based structural task decomposition methods like ALMA [46], HAMMER [38], MS-MARL [50], VAST [94] also exemplify this approach.

Several recent studies introduce automatic task decomposition (ATD) methods (see Section 5.1) that treat organizing agents into subgroups as a learning problem. Rochico [57] introduces an *organization control module* that receives local observations of all agents and makes adaptive teaming decisions using graph theory algorithms. The organization control module primarily focuses on establishing connections (edges) between agents. Subsequently, the connected components can be interpreted as teams, i.e., subgroups of agents. Shao et al. introduce the Self-organized Group (SOG) [103] method that dynamically organizes agents into subgroups, with these subgroups being supervised by designated *conductors*. SOG employs a technique known as *conductor election* (CE) to choose a set of conductors from the agents at specific time intervals. These CE methods can generate conductors using various strategies, including random selection, dissimilarity maximization, or reinforcement learning to treat the selection as a learnable action. Agarwal et al. [1] and Liu et al. [60] further incorporate communication mechanisms to support adaptability to the dynamic subgroup composition. QSCAN [45] is a value factorization framework that explicitly models coordination within sub-teams of agents to improve learning and value function factorization. It introduces a hierarchical way to represent this coordination based on sub-team size, allowing individual agents to execute learned policies in a decentralized manner. The Consensus-oriented Strategy (CoS) [97] improves collaboration in large teams by focusing on both group and individual policies. CoS uses a Vector Quantized Group Consensus (VQGC) module to learn discrete, stable, and distinguishable "group consensus" embeddings, which represent shared objectives or sub-tasks for dynamically formed agent groups. CoS follows the FC-HP paradigm, where the higher-level Group Consensus Policy (GCP) uses a hypernetwork architecture that takes the group consensus embedding as input to generate group-level decisions, aiming for effective coordination and long-term objectives within the group. At the lower level, a Group-Guided Policy (GGP) uses the same group consensus embedding to augment an agent's local observation, guiding individual actions while maintaining alignment with the group's consensus. The Structural Relational Inference Actor-Critic (SRI-AC) [151] automatically infers dynamic, pairwise interaction relationships between agents using a variational autoencoder (VAE) and graph neural networks (GNN), rather than relying on predefined structures. This learned relational structure allows agents, through a graph attention network (GAT) in the critic, to selectively focus on relevant neighbors, improving coordination in complex tasks without imposing a fixed hierarchy. Inspired by social psychology, Mao et al. propose Neighborhood Cognitive Consistency (NCC) [69] that uses graph neural networks to aggregate information from neighboring agents and encourages agents to develop similar understandings of their local environment. NCC employs variational inference techniques to align the neighborhood-specific parts of their internal representations, fostering coordinated behavior without imposing an explicit hierarchical control structure. Role-based learning methods classify agents according to their specific *roles*. Agents assigned the same role exclusively exhibit certain behaviors that are distinct from those of agents in other roles. In the context of role-based MARL, agents sharing a role effectively form a subgroup within the larger multi-agent system. However, it is noteworthy that role-based MARL algorithms, such as ROMA [125] and RODE [126], also incorporate elements of temporal task decomposition. This integration allows subgroups to sequentially tackle a series of subtasks, ultimately addressing the complexities of long-horizon tasks. A more detailed discussion of role-based MARL is reserved for Section 5.1.

4.5 Attention Mechanisms and Transformer Architectures

Attention mechanisms have revolutionized various domains of machine learning by enabling models to selectively focus on relevant parts of input data and filter out noise [47]. Initially introduced for sequence modeling in natural language processing [119], attention allows networks to dynamically weight the importance of different input features. The fundamental concept behind attention is to compute a weighted sum of values, where the weights are determined by a *compatibility function* between queries and keys.

In the MARL context, attention mechanisms have become increasingly prominent architectural components to address several core challenges, including scalability [47], effective communication among agents [47], and partial observability. By focusing only on pertinent information, attention mechanisms potentially mitigate the scalability issue of large-scale multi-agent systems where interaction complexity grows exponentially as the number of agents increases. With attention mechanisms, agents can dynamically attend to other most relevant agents or environmental features, reducing the complexity of the state space and the noise of environments. In addition, the attention weights provide insights into which agents or features are influencing a particular agent's decision-making process, enabling better interpretability of MARL models. This section reviews representative MARL methods that incorporate attention mechanisms as part of the architectural components.

In this section, we selectively review representative MARL methods that utilize attention mechanisms to enhance multi-agent communication. A recent survey by Hu et al. [44] for an extended discussion on how the attention mechanism is utilized in both single-agent and multi-agent reinforcement learning. This section specifically provides a deeper discussion on attention mechanisms in MARL and their scalability. We refer interested readers to the survey by Hu et al. for a broad discussion on attention mechanisms in reinforcement learning. Following the taxonomy proposed by Hu et al. [44], we classified literature with attention mechanisms in MARL into four main categories, namely, self-attention, graph attention, and multi-head attention.

4.5.1 Attention Mechanisms in MARL. Self-attention, especially in the form of transformer architectures, enables an agent to weigh the importance of different parts of its own observation or the information from other agents [47]. Self-attention computes attention scores between elements of the same set, typically using a *compatibility function* such as dot product, followed by softmax normalization. Self-attention mechanisms have been successfully integrated into several MARL frameworks to improve selective information processing and agent coordination. MAAC [47] implements self-attention mechanisms within critics of agents to dynamically weigh the importance of different agents' information, allowing each agent to selectively attend to relevant information from other agents. SparseMAAC [55] extends MAAC by implementing sparse attention weights, forcing agents to focus sharply on only the most critical agents in the environment, which accelerates convergence in complex scenarios. TarMAC [19] utilizes self-attention for targeted multi-agent communication, enabling agents to learn which agents to send messages to and how to interpret received messages. ATT-MADDPG [70] enhances centralized critics with an attention mechanism that employs a K-head architecture to model the joint policies of agents by adaptively weighting action conditional Q-values. DAACMP [71] employs dual attention mechanisms in an actor-critic architecture to adaptively select important messages via actor attention and process these messages via critic attention.

Graph attention networks (GATs) [120] extend attention mechanisms to graph-structured data, which is particularly suitable for multi-agent systems where interactions form a natural graph. By representing agents as nodes in a graph and their interactions as edges, graph attention mechanisms enable agents to selectively focus on the most relevant neighbors while filtering out less important information. One of the pioneering works in applying graph attention to MARL is MAGNet [68], which represents the multi-agent environment as a graph where agents and environmental objects are nodes. It employs self-attention to compute the relevance of other agents and objects, generating an environmental association graph. By using message-passing techniques on this graph, agents can more effectively process information about their surroundings and coordinate their actions,

leading to improved performance in complex multi-agent tasks. G2ANet [64] introduces a two-stage attention mechanism. In the first stage, the model determines which agents should interact with each other (hard attention), effectively constructing a dynamic interaction graph. In the second stage, it calculates the importance of each selected agent (soft attention), allowing agents to weigh the significance of information from different neighbors. GAMA [15] utilizes graph networks with attention mechanisms to facilitate information sharing between agents. By applying attention weights to the connections between agents, GAMA enables each agent to filter out irrelevant information and focus on critical data from the environment and other agents. Niu et al. [82] propose MAGIC, which combines graph neural networks with attention mechanisms to enable agents to learn which other agents to attend to and how to integrate their information, which leads to more effective teammate modeling and coordination in partially observable multi-agent environments. The IHA-MDGI [143] algorithm addresses the challenge of fusing heterogeneous information in multi-agent systems. By using heterogeneous graph attention, the model can effectively integrate information from diverse sources with different characteristics. A more recent development is DGAT-MACRL [134], which integrates distance-based graph attention into multi-agent systems. This approach calculates attention weights based on the physical distance between agents, allowing for more intuitive modeling of real-world interactions where proximity often correlates with relevance. The distance-based attention mechanism helps filter information from remote agents while focusing on nearby collaborators, making it particularly effective in communication-constrained environments.

Multi-head attention mechanisms have been increasingly adopted in MARL frameworks to enable more nuanced information processing and agent coordination. MA2DDPG [124] extends MADDPG by employing multi-head attention to extract and share key information between agents rather than raw observations, significantly improving convergence speed in UAV-assisted edge computing tasks. FT-Attn [35] incorporates multi-head attention to develop fault-tolerant communication policies that selectively identify valuable information even in highly noisy environments. MAATD3 [147] integrates multi-head attention with TD3 to optimize distributed control policies for energy systems, allowing agents to selectively incorporate relevant features while protecting entity privacy.

4.5.2 Transformer Architectures in MARL. Building upon the integration of attention mechanisms in MARL, a significant trend involves the use of Transformer architectures. These architectures, which are fundamentally based on attention mechanisms, have demonstrated remarkable capabilities in capturing complex dependencies within multi-agent systems. The Multi-Agent Transformer (MAT) [130] employs a Transformer with attention mechanisms within an encoder-decoder framework for joint policy optimization. This approach has achieved state-of-the-art performance in online MARL across various standard multi-agent environments, highlighting the power of Transformer networks in capturing complex multi-agent dynamics. Furthermore, methods like TransfQMix [31] utilize Transformer architectures to learn coordination graphs, showcasing the effectiveness of these attention-driven models in tackling complex multi-agent tasks. The success of these Transformer-based approaches underscores the power and versatility of attention mechanisms in advancing the field of multi-agent reinforcement learning.

Recently, the Stackelberg Decision Transformer (STEER) [148] addresses asynchronous action coordination in MAS as a Stackelberg game. It utilizes a dual Transformer architecture comprising an Inner Transformer Block (ITB) to process environmental state information and generate state embeddings, and an Outer Transformer Block (OTB) which autoregressively models the sequential, hierarchical decision-making process using masked multi-head self-attention to incorporate leader actions. The PDiT architecture [72] utilizes two cascaded Transformer modules with specialized functions. The first Transformer, the perceiver, focuses on environmental perception by processing observations at the patch level. The second Transformer, the decision-maker, concentrates on decision-making by conditioning on historical desired returns, the perceiver's outputs, and past actions. This design interleaves perception and decision-making blocks, allowing information exchange while maintaining

specialization within each block. PDiT is presented as generally applicable to various deep RL settings, including different algorithms (online/offline) and environments with diverse observation types (image, proprioception, hybrid image-language).

4.6 Mean Field Multi-Agent Reinforcement Learning

A significant challenge in scaling MARL arises from the curse of dimensionality, where the joint state-action space grows exponentially with the number of agents. Mean Field MARL (MFMARL) provides a powerful approximation framework to mitigate this issue. The core principle of MFMARL is to approximate the complex, high-dimensional interactions among numerous agents by modeling the interaction of a single representative agent with the average effect, or *mean field*, of the surrounding population. This simplification reduces the complexity of the learning problem, as an agent's policy or value function primarily depends on its own state and the aggregated information from the mean field, rather than the individual states and actions of all other agents.

Various MFMARL methods have been developed in recent years. Yang et al. introduce Mean Field Q-learning (MF-Q) [144], where agents learn Q-functions or policies by treating the actions of neighboring agents as drawn from an averaged distribution (the mean field), effectively decoupling the learning problem for each agent. Subramanian et al. extend mean field reinforcement learning by introducing multiple agent types, relaxing the core assumption that all agents are homogeneous and allowing for better modeling of diverse agent populations [32]. This algorithm handles scenarios where types are predefined or must be learned from observations, approximating complex interactions by considering the average effect of agents within each type. Bukharin et al. introduce ERNIE [9], a MARL framework that promotes policy smoothness (Lipschitz continuity) via adversarial regularization to enhance robustness against noisy observations, changing dynamics, and malicious actions. To improve training stability, the authors reformulate the adversarial regularization as a Stackelberg game. RoMFAC [154] is a robust mean-field actor-critic (MFAC) algorithm designed to counteract adversarial perturbations on agent states that can degrade performance in standard MFAC. It achieves robustness through a novel actor objective function that minimizes the action difference between clean and adversarial states, coupled with a repetitive regularization technique for this action loss.

By abstracting individual agent interactions into an aggregate effect, MFMARL offers a tractable and theoretically grounded approach for designing learning algorithms in large-scale multi-agent systems.

5 Towards Long-horizon Tasks: Challenges and Approaches

Real-world tasks often involve long decision-making horizons characterized by repeated sense-act cycles. In intelligent entities with lifespans, such as animals, these operations are indeed *lifelong*, where these entities engage in a virtually infinite sequence of lower-level decision or action steps. For example, humans possess a remarkable ability to formulate multi-step plans aimed at achieving long-term objectives [40].

A reasonable expectation from intelligent agents is proficiency in handling complex, long-horizon tasks and generalizing to new scenarios [40]. While many Multi-Agent Deep Reinforcement Learning (MADRL) algorithms have exhibited outstanding performance in short-horizon tasks, such as gaming [96] and robot manipulation [110], a direct implementation of these algorithms in long-horizon problems usually results in inferior performance. This challenge is more evident in scenarios where multiple sequential steps must be completed correctly to perform a task [40]. Practically, the long-horizon multi-agent learning becomes more challenging with realistic constraints such as *partial observability* and *sparse reward*, and the *temporal credit assignment* problem [2, 56] stands out as a primary challenge in reinforcement learning. In long-horizon MARL, crediting early or intermediate actions along a long trajectory for achieving a task is notably complex, particularly within environments characterized by *sparse rewards*, where rewards are only granted upon the final action. This issue also extends to the *structural credit assignment* problem [2], which arises in allocating credit among agents regarding their contributions.

The challenges mentioned above often result in MARL algorithms struggling to learn policies efficiently, necessitating a substantial number of environmental interactions–a pattern well-documented in the MARL literature. In such instances, a critical concern is *efficient exploration*. Agents must proficiently explore potentially rewarding states and actions, even if there is no immediate guiding feedback. Unlike the *temporal credit assignment* problem, which arises when the reward is only received at the final step, efficient exploration is a fundamental issue that arises from the beginning.

In the MARL literature, a universally accepted set of criteria for classifying what constitutes a long-horizon task has not yet been established. Therefore, it is imperative to foster a shared understanding of the current advancements in long-horizon MARL research. This section delivers a comprehensive review of the challenges that MARL algorithms encounter when dealing with long-horizon tasks and offers an overview of the state-of-the-art approaches for addressing those challenges.

Various MARL approaches have been proposed to address the challenge of learning action policies over long horizons. These approaches can be broadly categorized into two groups, namely the *subgoal-based* approaches and *exploration-based* approaches. *Subgoal-based* approaches decompose a long-horizon task into multiple subgoals that can each be achieved within a shorter temporal duration. This decomposition process can be applied recursively, resembling a "divide-and-conquer" strategy. The subgoals serve two purposes. Firstly, they form a temporally abstract decision-making process at the higher level since each subgoal is a higher-level decision or action that requires multiple time steps to achieve. Therefore, decision-making over such temporally abstract subgoals tackles the credit assignment problem at the higher level [39]. Secondly, subgoal-achievement can be associated with an intrinsic reward that eases the training of lower-level action policy even if the environmental rewards are sparse [51, 121]. *Exploration-based* approaches aim to address the long-horizon problem by direct learning with efficient exploration [56]. Such approaches are founded on the idea that enhancing an agent's ability to traverse the state space leads to more effective identification of states and actions that yield higher cumulative rewards.

5.1 Subgoal-Based Long-Horizon Multi-Agent Reinforcement Learning

The subgoal-based MARL approaches extend from the research on single-agent RL utilizing *subgoals* [7, 10, 20, 23, 36, 51, 91] and *intrinsically motivated* RL [18]. While many studies share a common definition and utilization of *subgoal*, they may interchangeably use other words to represent this concept, including "sub-goal," "sub-task" [20, 125, 126], "subtask," "subpolicy" [6] and "macro-action" [135]. In this work, we unify the representation of this concept as *subgoal* to avoid confusion.

Research on subgoal-based MARL typically seeks to address the following key questions [56]: (1) How to generate subgoals from the original task (*generation*); (2) How to assign appropriate subgoals to various agents (*selection*); (3) How to guide the agents in completing assigned subgoals (*completion*); (4) How to manage the termination of the selected subgoals and the activation of subsequent ones (*termination*). In this subsection, we review the work on subgoal-based MARL aimed at tackling the challenges posed by long-horizon tasks. We explore how the current research landscape addresses the generation, selection, completion, and termination of subgoals.

There are different approaches to perform subgoal-based MARL, such as subpolicy-oriented methods [6] or subgoal-oriented methods [56]. In this context, a *subgoal* refers to a desired terminal state at which the corresponding *subtask* is deemed to be completed. These approaches are collectively categorized under *temporal task decomposition,* characterized by the breakdown of each subtask from the overarching task. These methods stand in contrast to 'structural task decomposition' discussed in Section 4.4, which involves dividing a larger multi-agent system into smaller subgroups. This section explores both 'temporal task decomposition' methods and the 'structural task decomposition' methods that incorporate temporal task decomposition.

Rule-based task decomposition Earlier subgoal-based MARL methods took rule-based approaches to decompose tasks into a set of pre-defined subgoals. Tang et al. [116] introduced three multi-agent hierarchical reinforcement learning (MAHRL) algorithms: h-IL, h-Comm, and h-Qmix. These algorithms address the challenge of credit assignment in scenarios with sparse and delayed rewards by employing *temporal abstraction*. This technique decomposes the original task into multiple levels of subgoals. The authors handcraft two distinct temporal abstraction structures, each tailored to a specific task. For the *Multiagent Trash Collection* (MATC) tasks [67], the subgoals pertain to movements and trash collection. In contrast, for the *Fever Basketball Defense* (FBD) tasks [116], the subgoals revolve around the tactical movements of basketball players. Consequently, the construction of such subgoals may not be transferable across dissimilar tasks. Similarly, Xiao et al. [135] define subgoals as temporally extended actions, named *macro-actions*. To ensure a stable learning process of high-level policies of selecting macro-actions, the authors constrain the problem by assuming agents are provided with macro-action-conditioned policies.

Task decomposition with pre-defined rules raises a few practical issues worth considering. Firstly, manually pre-defining subgoals requires reliable prior domain knowledge of the dynamics of the environment and the nature of the long-horizon task [56, 126, 142]. As pre-defined subgoals are directly generated by analyzing the overall task, they commonly fail to transfer to other tasks that are not considered during the hard-coding process, limiting the generalizability of subgoal-based MARL approaches [56, 126]. In addition, it is also difficult to justify the quality of pre-defined subgoals. As subgoals are essentially important states of the overall gameplay trajectory that need to be achieved by agents, each state of the large state space could be a candidate for the pre-defined subgoal set. As a result, it is likely to select sub-optimal subgoals using hard-coded criteria. On the other hand, it is a common practice to rely on designers' intuition when determining the quantity of pre-defined subgoals. The number of subgoals may significantly affect the overall performance of MARL algorithms on long-horizon tasks. When decomposing a long-horizon task into various subgoals, each subgoal can be achieved by the agents within fewer steps compared to the original goal or task. However, large numbers of subgoals come with additional burdens, as coordinating multiple agents with large numbers of subgoals is challenging, especially in scaled-up large agent teams.

Automatic Task Decomposition (ATD) An alternative method for task decomposition is *automatic task decomposition* (ATD). ATD is a valuable tool for enabling MAS to function efficiently in various dynamic environments, without necessitating domain-specific knowledge. Depending on the target of decomposition, ATD methods can be classified into two categories, namely the *structural task decomposition* approach and the *temporal task decomposition* approach. The ATD methods that follow these approaches are summarized in Table 6 and reviewed as follows.

Role-based Methods. These methods comprise a kind of *dynamic grouping mechanism* where agents are grouped by roles [126]. In role-based MARL, the objective of learning is to learn policies for roles in a smaller observation-action space rather than for agents in an exponentially-increased state-action space. As the task is broken down into manageable components, the overall complexity of cooperative learning is reduced. This occurs because individual agents can concentrate on specific sub-problems, typically requiring smaller action-observation spaces. The potential for scalability is a crucial incentive for employing roles in multi-agent scenarios, where each role is linked to a particular, simplified task and its associated policy. Wang et al. propose the *role-oriented MARL* (ROMA) framework [125], which introduces dynamic and specialized roles to enable agents with similar responsibilities to share their learning. In ROMA, roles are represented as stochastic latent variables ρ_i drawn from Gaussian distributions parameterized by a *role encoder* neural network. The role encoder takes the agent's local observations as input, allowing roles to adapt to environment dynamics. The sampled role ρ_i is then fed into a *role decoder* to generate the parameters for the individual policy. RODE [126] extends ROMA into a bi-level MARL framework following the Flat Control, Hierarchical Policies (FC-HP) paradigm, as discussed in Section 3.3. RODE decomposes

Table 6. An overview of automatic task decomposition (ATD) methods of subgoal-based MARL

ATD Approach	Control Architecture and Policy Structure	Literature	Descriptions
Structural Task Decomposition	FC-FP	ROMA [125]	Encode the local observations of agents into a latent variable that represents roles or subtasks, and subsequently decode this latent variable to derive the individual policies of agents.
	FC-FP	Rochico [57]	Encode the local observation of agents into the edges, where agents are represented as nodes. Connected agents belong to the same subgroup.
	HC-FP	SOG [103]	Select some agents as conductors from agents. The conductor automatically organize all agents within its sight range.
	HC-FP	VAST [94]	Conduct a hierarchical value function factorization to factorize the overall value function into sub-groups of agents.
Temporal Task Decomposition	FC-HP	LDSA [142]	Learn a subtask encoder that uses vector representation for each subtask.
	FC-HP	MASER [48]	Generate subgoals for each agent by selecting states from the experience replay buffer.
	FC-HP	Nguyen et al. [80]	Propose a new mixing network architecture that can learn to assign credit to emergent roles, allows for transferring pre-trained models across different team sizes.
	FC-HP	RODE [126]	Learns action representations based on their effects and clusters actions into subsets. Treats each cluster of actions as a distinct subtask.
	HC-FP	SAMA [56]	Uses pre-trained language models (PLMs) to provides task decomposition and allocation for efficient MARL.

each full task into a set of *sub-tasks* with pre-training, each having a smaller observation-action space that is shared by a group of agents with the same role. RODE essentially learns action representations based on their effects and clusters actions into subsets. It then treats each cluster of actions as a distinct subtask, with each subtask focused on fulfilling the functionality of a specific subset of actions. The higher level of RODE is a *role selector*, which coordinates role assignments in a smaller role space and at a lower temporal resolution. At the lower level, role policies explore strategies in reduced observation-action spaces. While RODE outperforms various state-of-the-art MARL algorithms on the benchmark SMAC tasks [98], Yang et al. [142] show that RODE may fail to solve subtasks when some basic actions are necessary for all subtasks, such as the movement actions in SMAC. Moreover, the effect of each action dynamically changes with the environment and the task. Therefore, it may be challenging to determine the actions' effect solely through pre-training. To address these limitations, Yang et al. [142] propose a framework called LDSA to learn dynamic subtask assignments in cooperative MARL. LDSA first constructs a vector representation for each subtask using a subtask encoder. Agents then select subtasks based on the cosine similarity between their observation-action history embeddings and subtask representations. This allows dynamic assignment of agents to subtasks based on their abilities. Agents assigned to the same subtask share policy parameters and experience, while different subtasks have distinct policies associated with their representations via a *subtask decoder*, which balances training complexity and behavior diversity. Jeon et al. [48] propose a method for role assignments named *MARL with subgoals generated from experience replay*

buffer (MASER) to automatically generate subgoals for each agent by selecting states from the experience replay buffer that maximize a weighted combination of the agent's local Q-value and the global Q-value. *Intrinsic rewards* are designed to encourage agents to reach their subgoals while maximizing the joint action value. The intrinsic rewards are based on the distance between the current state and the subgoal state after an actionable representational transform. Nguyen et al. [80] propose a new mixing network architecture that can learn to assign credit to emergent roles and agents playing those roles. This allows for transferring pre-trained models across different team sizes. The authors enable *curriculum learning* by first pre-training on a small team where exploration is cheaper, then transferring the model to continue training on a larger team.

Methods using Language Models. Large language models can also be used as automatic subtask generators for agents that follow natural language instructions. Li et al. [56] introduce a method named *Semantically Aligned Task Decomposition in MARL* (SAMA), employing *pre-trained language models* (PLMs) for the task decomposition and subtasks allocation. Following a bi-level HC-FP control paradigm (detailed in Section 3.4), SAMA's higher-level controller, modeled as the *gpt-3.5-turbo*, oversees task generation, task decomposition, and subtask allocation. Concurrently, the lower-level agents accomplish these subtasks with pre-trained goal-conditioned policies, taking natural language inputs and carrying out primitive actions within the environment. The higher-level agent is prompted with few-shot, *in-context exemplars* via the chain-of-thought (CoT) [128] paradigm. Each in-context exemplar comprises the task manual, the extant environmental state, the current local observations of all lower-level agents, the generated tasks, the decomposed subtasks or allocation outcomes, and the corresponding reasoning for each of the above elements. At each time step t, the higher-level controller agent generates a semantically aligned *subgoal* g_t that necessitates cooperative achievement by low-level agents. Subsequently, the higher-level agent decomposes g_t into N distinct sub-subgoals: $g_t^1, g_t^2, \ldots g_t^N$, with each designated for an individual lower-level agent. Thereafter, lower-level agents complete the allocated sub-subgoals by utilizing pre-trained goal-conditioned language-grounded RL models.

Multi-task Learning. It is natural to draw a connection between the *task decomposition* approach and *multi-task learning*. The former focuses on disassembling and tackling a comprehensive task, while the latter explores how agents can effectively handle multiple tasks. For instance, Andreas et al. [6] introduce a framework for multi-task deep reinforcement learning that is guided by *policy sketches*. We shall elaborate on the potential of multi-task MADRL later in the discussion in Section 7.

5.2 Efficient Exploration for Long-Horizon Sparse-Reward MARL

In contrast to the task decomposition strategies discussed in the previous section, a distinct class of approaches for addressing the challenges of long-horizon sparse-reward MARL involves direct exploration of the state, action, or policy space to uncover rewarding states and behaviors. These methods offer the advantage of bypassing the need for subgoal generation or pre-defined task decomposition, which can introduce overhead. However, it is important to note that this approach may entail longer training times as agents explore a sparse reward environment from scratch. Notably, prior research by Nachum et al. [79] demonstrated the potential benefits of task decomposition hierarchies in enhancing exploration. Despite the trade-offs, exploration-based MARL approaches remain valuable, and in this section, we delve into a selection of significant methods within this category. Several distinct categories of approaches have emerged to address the challenge of long-horizon sparse-reward MARL using efficient exploration, discussed as follows. Table 7 summarizes the method for efficient exploration for long-horizon sparse-reward MARL.

Subspace Exploration. Here, a subspace refers to a lower-dimensional projection of the original joint state space, which is used since the original joint state space grows exponentially with the number of agents, making exploration challenging. CMAE (Counter-based MARL for Subspace Exploration) [62] stands out as one such

Table 7. An overview of the methods for efficient exploration for long-horizon sparse-reward MARL.

Category	Approach	Key Features	Tasks	Performance
Subspace Exploration	CMAE [62]	Uses entropy-based technique for goal selection. Focuses on low-dimensional space and uses count-based strategy to prioritize under-explored areas.	SMAC	Outperforms QMIX on SMAC.
	SAME [137]	Entropic exploration objective. Algorithm to improve exploration objective's lower bound. Scales linearly with agent number.	SMAC and GRF	Outperforms QMIX and CMAE on hard SMAC scenarios and GRF.
Joint Policy Diversity	Xu et al. [136]	Policy-level diversity. "Constrained joint policy diversity" measure. Encourages diverse strategy exploration.	MPE, GRF, and SMAC	Outperforms QMIX and CMAE on MPE, GRF, and SMAC.
Curiosity-Driven Exploration	MAVEN [66]	Latent space for hierarchical control Value-based agents conditioned on shared latent variable.	SMAC	Outperforms QMIX on SMAC.
	EMC [152]	Balances centralized and decentralized curiosity. Utilizes Q-value prediction errors as intrinsic rewards. Integrates episodic memory for experience utilization.	SMAC	Outperforms MAVEN on hard SMAC scenarios.

approach. Specifically, it starts by selecting shared exploration goals using an entropy-based technique to identify valuable states for exploration. Agents then work together in a coordinated fashion to reach these goals, optimizing exploration efficiency. CMAE also explores a low-dimensional restricted space within the state space and employs a count-based strategy to prioritize under-explored areas. This method outperforms QMIX [96] and its variants in SMAC tasks. In the same category, SAME (Subspace-Aware Multi-agent Exploration) [137] introduces an entropic exploration objective to encourage agents to explore sub-state spaces with higher uncertainty. However, SAME is sensitive to the number of agents and involves estimating state distributions in high-dimensional sub-state spaces, which can be impractical for many tasks. To address these challenges, SAME also includes an algorithm to improve the exploration objective's lower bound, reducing computational complexity. This modified approach ensures that computational cost scales linearly with the number of agents and simplifies distribution estimation for practical, real-world applications. SAME outperforms QMIX and CMAE in a few hard SMAC scenarios and GRF. In both CMAE and SAME, the subspace is defined based on the input dimensions that have the most effect on the joint reward.

Joint Policy Diversity. Xu et al. [136] developed an approach incorporating policy-level diversity and a novel "constrained joint policy diversity" metric to enhance agent exploration in sparse-reward environments. When tested on challenging benchmarks including MPE, GRF, and SMAC, their method substantially outperformed leading approaches like QMIX and CMAE across most tasks. Notably, in certain scenarios, it was the first to successfully develop winning strategies without domain-specific knowledge under sparse-reward conditions. The approach proved particularly effective by exploring significantly more states than traditional exploration methods in reward-free settings.

Curiosity-driven Exploration. A representative method in this category is MAVEN (Multi-agent Variational Exploration) [66], which addresses efficient exploration in cooperative deep MARL, particularly in centralized training with decentralized execution settings. It introduces a latent space for hierarchical control, enabling value-based agents to condition their actions on a shared latent variable governed by a hierarchical policy. This facilitates committed, temporally extended exploration while respecting representational constraints. Evaluated on the SMAC domain, MAVEN demonstrates significant performance improvements over QMIX in complex multi-agent tasks. Another approach, EMC (Episodic Multi-agent Reinforcement Learning with Curiosity-driven Exploration) [152], addresses efficient exploration and coordination in deep MARL by balancing centralized and decentralized curiosity through Q-value prediction errors as intrinsic rewards. These prediction errors capture

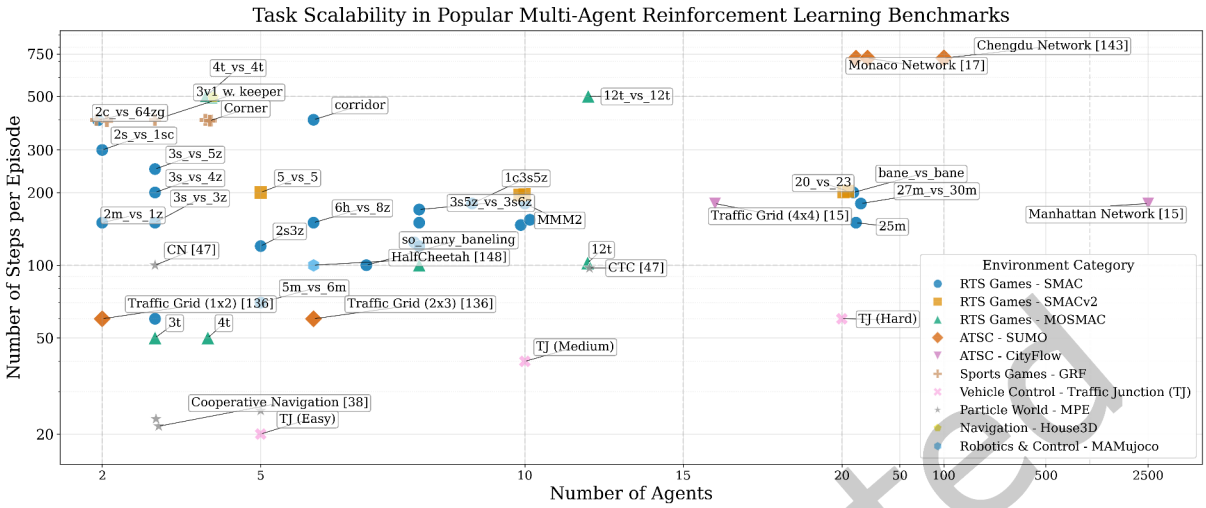

Fig. 3. The scale of tasks in MARL environments discussed in this survey. Note that the scale of tasks in some environments (e.g., MPE) is highly customizable, with implemented settings varying across the literature. For these environments, we specifically label papers that report their exact scale settings. Some tasks and environments discussed in this paper are not shown in this figure due to insufficient information about their temporal horizon.

both novelty and inter-agent dependencies, guiding agents toward promising states. EMC further enhances experience utilization by integrating episodic memory, enabling efficient storage and retrieval of past successful trajectories.

6 Environments and Benchmarks

Environments and benchmarks constitute foundational elements in the systematic study of multi-agent reinforcement learning. The environment architecture fundamentally determines the state and action spaces accessible to agents and establishes the probabilistic framework for state transitions. Therefore, the methodical selection of appropriate tasks for algorithm development and evaluation represents a critical decision point in MARL research.

The prohibitive computational and financial costs associated with training MARL models in real-world settings necessitate the utilization of simulated environments. Certain simulation frameworks have emerged as de facto standards within the research community, subsequently facilitating the development of standardized benchmarks derived from these canonical environments. The StarCraft II platform exemplifies this phenomenon, serving as the foundation for influential benchmark suites such as SMAC [98] and its successor, SMACv2 [25]. These benchmarks, while originating from a common environmental substrate, are characterized by distinct task properties and complexity gradients, thereby providing a rich spectrum of challenges for comprehensive MARL evaluation.

A persistent challenge in MARL research, however, is establishing consistent evaluation protocols. As new state-of-the-art algorithms emerge, researchers often employ different benchmarks within the same environment, select diverse scenarios within a benchmark, or report results using inconsistent metrics. These methodological variations introduce significant complexities that impede direct and fair comparisons between models. This section reviews prevalent environments and benchmarks in contemporary MARL research, with the scale of common tasks and environments summarized in Figure 3. We highlight a significant gap in the field: despite the availability of customizable scenarios, there remains a dearth of benchmarks specifically designed for scaling

MARL systems. Table 8 summarizes the scale of common tasks and environments employed by MARL methods discussed in this survey.

6.1 Real-Time Strategy Game Environments

Video games are suitable testbeds for MARL algorithms as they typically provide explicit winning conditions and defined state and action spaces. Additionally, agent behaviors can be compared with those of human players, enabling intuitive evaluation of learned policies. Furthermore, these behaviors can be analyzed and summarized as strategies.

In recent years, Real-Time Strategy (RTS) games have garnered significant interest within the MARL community. This interest primarily stems from the natural alignment of RTS games with multi-agent environments, where each agent independently controls a unit. A salient advantage of RTS games is their intrinsic flexibility, which facilitates the dynamic modification of the number of agents and eases the creation of a variety of scenarios. Consequently, RTS games have become widely recognized as experimental testbeds for exploring complex environments in MARL, characterized by diverse team compositions of agents and varying planning horizons.

6.1.1 StarCraft: Broodwar (SCBW). SCBW served as a prominent experimental platform for MARL research prior to the introduction of the StarCraft II API. Singh et al. [106] utilized SCBW as their simulation environment to introduce StarCraft-exploration tasks, which subsequently influenced numerous follow-up investigations in StarCraft II. Foerster et al. [28] also employed SCBW in their work, specifically leveraging TorchCraft [113] as their implementation framework. These early implementations established fundamental approaches for multi-agent coordination in complex, partially observable environments that continue to inform contemporary research directions.

6.1.2 StarCraft II. StarCraft II [122] has emerged as a prominent simulation environment extensively utilized by the MARL research community. Researchers have pursued various tasks featuring different characteristics of multi-agent systems and scenarios. Some studies focus on simulating challenging combat scenarios between two confronted unit teams [25, 98] to evaluate MARL strategies on micro-management tasks, while some others aim at enhancing the exploration efficiency of MARL algorithms [49]. Within this domain, three primary sets of benchmark tasks have been established using StarCraft II: SMAC [98], SMACv2 [25], and SMAC-Exp [49]. Notably, SMACv2 and SMAC-Exp are extensions of the original SMAC benchmark, each introducing unique challenges pertinent to MARL research.

Similar to SCBW, SMAC [98] challenges various MARL algorithms [30, 92, 96] with a set of micro-management tasks. SMAC provides agents with a partially observable environment, which is the typical setting for StarCraft II experiments. As a result, agents observe their local information and receive a global reward. SMACv2 [25] addresses the limitations of the original benchmark, SMAC, by introducing procedural content generation (PCG) to increase stochasticity. SMACv2 randomly generates team compositions and agent start positions for each episode, requiring agents to learn to cooperate in a diverse range of scenarios. The benchmark also updates the sight and attack ranges to increase unit diversity. State-of-the-art algorithms struggle with many SMACv2 scenarios, indicating that SMACv2 poses substantial new challenges. SMACv2 combines partial observability, complex dynamics, and high-dimensional observation spaces, making it a comprehensive testbed for cooperative MARL methods.

The StarCraft Multi-Agent Exploration Challenges (SMAC-Exp) [49] is another extension of the SMAC benchmark. SMAC-Exp introduces multi-stage tasks and environmental factors that agents must learn to accomplish. The defensive scenarios in SMAC-Exp involve cooperative decision-making among agents using units like Marines, Marauders, and Tanks. Agents must leverage environmental factors such as topography and unit combinations to defeat opponents. The offensive scenarios in SMAC-Exp require agents to accomplish goals, such as progressively

locating and defeating adversaries. Agents need to explore the map, locate enemies, and effectively use their troops to defeat adversaries. The state and observation features in SMAC-Exp include additional information on terrain levels, such as the pathing grid and terrain height. Agents must distinguish whether opponents are located on higher ground or not in these scenarios. The action space includes basic actions, the number of enemies, and the number of neutral buildings. Agents can also utilize unit skills, such as changing a general tank to siege mode. The authors evaluated 11 MARL algorithms on the SMAC-Exp benchmark using sequential and parallel episodic buffers.

6.2 Other Game-Based Environments

6.2.1 Overcooked. The Overcooked [13] environment is a cooperative multi-agent game where players control chefs in a kitchen and work together to prepare meals. It has been used in a recent study of subgoal-based MARL, SAMA [56], to evaluate language-grounded RL techniques. The Overcooked environment presents significant coordination challenges for human players. The goal is to place three onions in a pot, take out the resulting soup on a plate, and deliver it as many times as possible within a time limit. The environment consists of a human (H) who is close to a dish dispenser and some cooked soup, and an AI agent (AI) who is facing a pot that is not yet full. The optimal strategy is for H to put an onion in the partially full pot and for AI to put the existing soup in a dish and deliver it. However, coordination failures can occur if AI expects H to be optimal and H plans to pick up a plate to deliver the soup.

6.2.2 Smallville. The Smallville [89] environment is a sandbox world comprising various areas such as houses, cafes, stores, and parks. Each area is represented as a node in a subgraph, with leaf nodes representing objects within those areas. The sandbox world is inhabited by a community of 25 unique generative agents, each modeled as a large language model (LLM), namely ChatGPT (GPT-3.5-turbo) [83]. These agents have identities, occupations, and relationships with other agents, which are stored as seed memories. The agents interact with the world and each other through natural language communication. They can perform actions translated into concrete movements in the sandbox world, displayed as emojis above their avatars. Users and agents can influence the state of objects in the Smallville environment, and the agents' behaviors evolve as they interact with each other and the world.

6.2.3 Neural MMO. Neural MMO [108] aims to capture properties of survival and competition in nature by simulating a massively multiplayer online role-playing game (MMORPG) environment. In this environment, each agent is represented by a neural network that learns to survive using deep reinforcement learning. The paper demonstrates through experiments that large populations of agents in the simulation act as competitive pressure, encouraging exploration of the environment and the development of skillful behavior. Additionally, when agents are organized into species with shared policy parameters, each species naturally diverges to occupy its behavioral niche.

6.3 Multi-Agent Particle Environment (MPE)

The multi-agent particle environment (MPE) originated from the particle world environment introduced by Mordatch et al. [78] and was further extended by Lowe et al. [65]. MPE contains various tasks, including cooperative communication, navigation, keep-away, physical deception, predator-prey, and covert communication. In the cooperative communication environment, there are two cooperative agents, a speaker and a listener, who navigate to landmarks based on communication. In the cooperative navigation environment, agents cooperate to reach landmarks while avoiding collisions. In the keep-away environment, cooperating agents try to reach a target landmark while adversarial agents try to prevent them. In the physical deception environment, cooperating agents spread out to deceive an adversary and reach a target landmark. In the predator-prey environment,

Table 8. The scale of MARL tasks that were experimented with the MARL methods reviewed in this survey. Note that the scale of tasks in some environments (e.g., MPE) is highly customizable, with implemented settings varying across the literature. For these environments, we specifically label papers that report their exact scale settings.

Environment Category	Environment/Task	Scalability		Representative Methods
		Agent Team Sizes	Time Horizons	
RTS Games	StarCraft Broodwar	2 [30] - 20 [92]	Not specified	[28, 30, 92, 106, 117]
	StarCraft II (SMAC)	2 - 27	60 - 400	[30, 45, 48, 96, 100, 126]
	StarCraft II (SMACv2)	5 - 20	200	[96, 123, 145]
	StarCraft II (SMAC-Exp)	5 - 15	120	[96, 111, 115]
	StarCraft II (MOSMAC)	3 - 12 [33]	50 - 500	[33, 96, 115]
Other Game-based Environments	Google Research Football (GRF)	2 - 11	400 - 3000	[82, 130, 138, 139, 148]
	Overcooked	2 - 3	400 [56]	[56, 135]
Adaptive Traffic Signal Control (ATSC)	CityFlow	16 [14], 2510 [14]	180 [14]	[14]
	SUMO	2 [133] - 100 [143]	60 [133] - 720 [143]	[17, 133, 143]
Robotics & Control	MAMuJoCo	6 [148]- 8 [148]	100 [148]	[139, 148]
	Bi-DexHands	2 [130]	Not specified	[130]
Particle Environments	Multi-agent Particle Environments (MPE)	2 [65] - 11 [3]	25 [38] - 100 [47]	[15, 38, 47, 65]
Vehicle Control	Traffic Junction (Grid)	5 [19] - 20 [106]	20 [19] - 60 [106]	[17, 19, 106]
In-house Navigation	House3D	4 [19]	500 [19]	[19]

cooperating agents chase an adversary while avoiding obstacles. In the covert communication environment, a speaker agent communicates a message to a listener agent while an adversarial agent tries to intercept the message. Actions in MPE can be discrete or continuous.

6.4 Transportation Simulated Environments

6.4.1 CityFlow. The advancement of adaptive traffic signal control (ATSC) algorithms requires robust simulation environments capable of handling large-scale urban traffic networks. CityFlow [149] serves as both a high-performance traffic simulator and a specialized MARL environment for traffic signal optimization. It offers computational efficiency for large networks, scalability supporting thousands of intersections and vehicles, and flexible configuration for both synthetic and real-world traffic data. CityFlow models individual vehicle dynamics at each time step through efficient multi-threading, optimized data structures, and streamlined movement algorithms; it also represents each traffic signal as an independent agent, supporting various MARL architectures (centralized, decentralized, and hybrid). The simulator provides standardized observation spaces, action spaces, and reward functions, generating thousands of state-action-reward samples per episode for effective training of deep reinforcement learning models in traffic management.

6.4.2 Flow. FLOW [132] provides a modular framework that allows for the creation, study, and control of complex traffic scenarios. It enables the composition of diverse mixed autonomy traffic scenarios for study with deep reinforcement learning. Different from the CityFlow [149] environment, where the focus is on traffic signal control, the scenarios in FLOW are composed of different modules, including the network module, which specifies the physical road layout, and the actors module, which describes the physical agents in the environment and their control interfaces. Other modules include the observer module, which maps the state to observations for the actors; the control laws module, which dictates the behaviors of the actors; and the dynamics module, which describes different aspects of the system evolution, such as vehicle routes, demands, and traffic rules. Agents in FLOW can take actions based on the observations provided by the observer module, and the control laws module determines these actions.

7 Outstanding Challenges and Promising Directions

Previous sections have explored MARL methods that scale toward larger agent teams and longer time horizons. Despite significant progress, scaling MARL systems continues to present fundamental challenges. These include: exponential computational complexity as teams expand; persistent non-stationarity that destabilizes learning; poor sample efficiency, particularly in real-world domains; coordination difficulties under partial observability; task decomposition for long-horizon problems; limited adaptation to dynamic environments; and challenges in accurately modeling multi-agent dynamics. The following subsections explore promising directions that address these interrelated challenges.

7.1 Efficient Multi-Agent Reinforcement Learning via Policy Initialization and Foundation Models

Scaling up MARL introduces notably complex challenges, as discussed in Sections 4 and 5. An effective strategy to tackle these complexities involves leveraging prior knowledge, such as integrating pre-trained policies or foundation models into the learning process. This approach can expedite training, mitigate the pervasive issue of non-stationarity, and ultimately reduce the complexity and cost of learning.

Initializing policies, potentially at different levels within hierarchical frameworks (such as the HC-FP or FC-HP), can significantly speed up learning by reducing uncertainty in pre-trained components. Policy initialization has been explored in several preliminary studies, where the low-level controllers benefit from pre-trained high-level policies, as the initialized policies could be viewed as part of the environment from the perspective of new agents trained from scratch. For instance, HSD [141] demonstrated that incorporating agents initialized with expert knowledge (scripted bots or fixed skills) improved performance compared to training entirely from scratch, although this might not hold for flat MARL algorithms like QMIX [96] or IQL [77, 115]. Similar approaches were also adopted in FCRL [52] and HiSOMA [33]. Furthermore, pre-training in simulation offers a cost-effective way to bootstrap MARL systems, particularly for real-world applications like robotics, where data collection is expensive, as shown by RLS [43]. More recently, the integration of large pre-trained models, often termed *foundation models*, presents a powerful avenue. These models, trained on vast datasets, exhibit remarkable adaptability. Their application in MARL, such as using Pre-trained Language Models (PLMs) for automatic subtask generation (e.g., SAMA [56]) as discussed in Section 5, holds significant potential, especially for complex task decomposition in long-horizon problems. While hybrid learning — applying different paradigms (e.g., centralized, decentralized) across hierarchical levels — is also feasible, the escalating computational demands of large-scale systems suggest that leveraging pre-initialized policies and foundation models will become increasingly crucial for efficient scaling.

7.2 Model-Based Multi-Agent Reinforcement Learning

While this paper predominantly discusses *model-free* multi-agent reinforcement learning (MARL) methods, which operate without estimating the transition probability distribution and reward function in the POMDP framework, there is growing interest in *model-based* MARL approaches [24, 139]. These emerging methods involve learning a model of the environment through agents' interactions with it, aiming to understand environmental dynamics and use this understanding to guide decision-making. Although model-based methods are often more sample-efficient than their model-free counterparts, they can struggle to learn accurate state representations. Despite these challenges, model-based MARL remains a promising yet under-explored area of research.

Some recent studies have begun to address this gap. Xu et al. [139] introduce *Model-Based Value Decomposition* (MBVD), a model-based MARL algorithm that allows agents to interact with a learned virtual environment, evaluating the current state value based on imagined future states in the latent space, effectively providing agents with foresight. MAMBA (Multi-Agent Model-Based Approach) [24] reduces the need for extensive environmental interactions by utilizing *imaginary rollouts* based on a learned world model. It also incorporates

discrete communication protocols for efficient inter-agent communication, supporting decentralized execution and allowing agents to make independent decisions based on local observations and limited-bandwidth communication. These methods, generalizable across various MARL frameworks, highlight the potential of model-based MARL as a fertile ground for future research.

7.3 Multi-Agent Meta Reinforcement Learning

While multi-agent environments are typically non-stationary, agents in these settings can be perceived as engaging with a series of stationary phases, each characterized by a specific, limited timescale. Within these distinct intervals, the state-transition probabilities remain consistent, but they vary when the environment transitions to the next stationary phase. The multi-agent learning problem can thereby be re-constructed as a *continuous adaptation* [4] problem. Such reinforcement learning problems could be addressed by *meta-learning* [27] approaches, where agents are trained not only to perform specific tasks but also to quickly adapt to new, changing environments or tasks. As the non-stationarity issue is one of the main challenges faced by structurally scaling up MARL (see Section 4), meta-learning is a promising research direction. Moreover, modeling the training process of multi-agent agents in a sequential task aligns with the concept of multi-task learning and temporal abstraction (see Section 5). Notably, Al-Shedivat et al. [4] introduce a single-agent few-shot gradient-based meta-learning approach, built upon the Model-Agnostic Meta-Learning (MAML) framework [26], where agents learn to exploit dependencies between successive tasks and generalize to co-adapting adversarial agents at test time. While several studies, including the surveys by Papoudakis et al. [86] and Gronauer and Diepold [34], have classified this method as a MARL approach, the essence of this problem is still a single-agent setting. Several recent studies have extended MARL methods with meta-learning, a class of approaches known as *multi-agent meta reinforcement learning*. Mao et al. [73] introduce a theoretical framework for multi-agent meta reinforcement learning for analyzing Nash equilibria in two-player zero-sum Markov games, Markov potential games, and coarse correlated equilibria in general-sum Markov games. The authors propose several MARL algorithms, featuring optimistic policy mirror descents and stage-based value updates, ensuring near-optimal performance in suboptimal initial conditions. Yun et al. [146] present a method named Quantum Multi-Agent Meta Reinforcement Learning (QM2ARL), which is built upon Quantum Neural Networks (QNNs). QM2ARL exploits two types of trainable parameters—angle parameters and pole parameters—applying angle training for meta-QNN learning and pole training for few-shot or local-QNN learning. Despite the insights provided by existing studies, multi-agent meta-reinforcement learning approaches specifically tailored for scaling up MARL remain an under-explored area in the field.

8 Conclusion

This survey has comprehensively analyzed the landscape of multi-agent reinforcement learning (MARL) algorithms across two critical dimensions: larger cooperative agent teams and longer temporal horizons. Our central contribution is a novel taxonomy categorizing MARL frameworks along two orthogonal dimensions: **external control architectures among agents** and **internal policy structures**, yielding four paradigms: FC-FP, HC-FP, FC-HP, and HC-HP. Our analysis reveals that most contemporary MARL research has focused on the FC-FP paradigm, with significant progress since 2015. However, scaling up MARL introduces formidable challenges: larger agent teams must contend with *non-stationarity* and *the curse of dimensionality*, while longer temporal horizons intensify *credit assignment* problems. Simultaneously scaling in both dimensions compounds these challenges, necessitating novel approaches.

Abstraction emerges as a unifying principle for addressing these challenges through task decomposition. *Structural task decomposition* (HC-FP paradigm) coordinates large-scale agents through hierarchical team organization, while *temporal abstraction* (FC-HP paradigm) decomposes long-horizon tasks into tractable subtasks. Our research indicates that these decomposition approaches represent promising avenues for scaling MARL systems, with the

HC-HP paradigm offering particular potential for simultaneously addressing both scaling dimensions, despite being currently underexplored in the literature. Our survey also highlights a critical shortage of appropriate benchmarks, with even popular platforms like SMAC limited to fewer than 30 agents with a maximum of 400 actions per episode. This underscores the need for more challenging environments and standardized evaluation criteria that better reflect real-world scaling requirements.

Future research directions include exploring connections with *multi-task learning*, leveraging *imitation learning* to bypass expensive exploration, integrating *multi-objective learning* for complex long-horizon scenarios, and developing methods that accommodate dynamic agent populations. For readers seeking a deeper understanding of these topics, we recommend several foundational resources. The theoretical underpinnings of multi-agent systems and reinforcement learning are comprehensively covered in works by Shoham and Leyton-Brown [104] and Sutton and Barto [112], respectively. Plaat's work [95] bridges classical RL with modern deep learning approaches, while Albrecht et al. [5] provide a comprehensive introduction specifically to MARL. Online educational resources from Stanford (CS234), UC Berkeley (CS285), and OpenAI's Spinning Up project offer structured learning paths of MARL.

This survey serves as a roadmap for researchers navigating the complex landscape of scaling multi-agent reinforcement learning, highlighting both promising directions and critical challenges that must be addressed to realize the full potential of MARL in complex real-world applications.

Acknowledgments

This research was conducted in collaboration with DSO National Laboratories, Singapore, and supported in part by the Presidential Doctoral Fellowship in Computing awarded to Minghong Geng and the Lee Kong Chian Professorship awarded to Ah-Hwee Tan, both by Singapore Management University.

References

[1] Akshat Agarwal, Sumit Kumar, Katia Sycara, and Michael Lewis. 2020. Learning Transferable Cooperative Behavior in Multi-Agent Teams. In *Proc. of the 19th International Conference on Autonomous Agents and MultiAgent Systems* (Auckland, New Zealand) *(AAMAS '20)*. IFAAMAS, Richland, SC, USA, 1741–1743. https://dl.acm.org/doi/10.5555/3398761.3398967

[2] Adrian K. Agogino and Kagan Tumer. 2004. Unifying Temporal and Structural Credit Assignment Problems. In *Proc. of the Third International Joint Conference on Autonomous Agents and Multiagent Systems* (New York, NY, USA) *(AAMAS '04)*. IEEE Computer Society, Los Alamitos, CA, USA, 980–987. doi:10.1109/AAMAS.2004.10098

[3] Sanjeevan Ahilan and Peter Dayan. 2019. Feudal Multi-Agent Hierarchies for Cooperative Reinforcement Learning. arXiv:1901.08492v1 [cs.MA]

[4] Maruan Al-Shedivat, Trapit Bansal, Yuri Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. 2018. Continuous Adaptation via Meta-Learning in Nonstationary and Competitive Environments. In *International Conference on Learning Representations* (Vancouver, Canada). https://openreview.net/forum?id=Sk2u1g-0-

[5] Stefano V. Albrecht, Filippos Christianos, and Lukas Schäfer. 2024. *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*. The MIT Press, Cambridge, Massachusetts. https://www.marl-book.com

[6] Jacob Andreas, Dan Klein, and Sergey Levine. 2017. Modular Multitask Reinforcement Learning with Policy Sketches. In *Proc. of the 34th International Conference on Machine Learning* (Sydney, Australia) *(Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 166–175. https://proceedings.mlr.press/v70/andreas17a.html

[7] Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. 2016. Unifying Count-Based Exploration and Intrinsic Motivation. In *Advances in Neural Information Processing Systems* (Barcelona, Spain), Vol. 29. Curran Associates, Inc., Red Hook, NY, USA, 1471–1479. https://proceedings.neurips.cc/paper_files/paper/2016/file/afda332245e2af431fb7b672a68b659d-Paper.pdf

[8] Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. 2002. The Complexity of Decentralized Control of Markov Decision Processes. *Mathematics of OR* 27, 4 (Nov. 2002), 819–840. doi:10.1287/moor.27.4.819.297

[9] Alexander Bukharin, Yan Li, Yue Yu, Qingru Zhang, Zhehui Chen, Simiao Zuo, Chao Zhang, Songan Zhang, and Tuo Zhao. 2023. Robust Multi-Agent Reinforcement Learning via Adversarial Regularization: Theoretical Foundation and Stable Algorithms. In *Advances in Neural Information Processing Systems* (New Orleans, LA, USA), Vol. 36. Curran Associates, Inc., Red Hook, NY, USA, 68121–68133. https://proceedings.neurips.cc/paper_files/paper/2023/file/d6f8517fceeca1e2cd61721dff786c14-Paper-Conference.pdf

[10] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. 2019. Exploration by Random Network Distillation. In *International Conference on Learning Representations* (New Orleans, Louisiana, USA). https://openreview.net/forum?id=H1lJJnR5Ym

[11] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. 2008. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38, 2 (March 2008), 156–172. doi:10.1109/TSMCC.2007.913919

[12] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. 2010. *Multi-Agent Reinforcement Learning: An Overview*. Studies in Computational Intelligence, Vol. 310. Springer, Berlin, Heidelberg, 183–221. doi:10.1007/978-3-642-14435-6_7

[13] Micah Carroll, Rohin Shah, Mark K. Ho, Thomas L. Griffiths, Sanjit A. Seshia, Pieter Abbeel, and Anca Dragan. 2019. On the Utility of Learning about Humans for Human-AI Coordination. In *Advances in Neural Information Processing Systems* (Vancouver, Canada), Vol. 32. Curran Associates Inc., Red Hook, NY, USA, 5174–5185. https://proceedings.neurips.cc/paper_files/paper/2019/file/f5b1b89d98b7286673128a5fb112cb9a-Paper.pdf

[14] Chacha Chen, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui Li. 2020. Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *Proc. of the AAAI Conference on Artificial Intelligence* 34, 04 (April 2020), 3414–3421. doi:10.1609/aaai.v34i04.5744

[15] Haoqiang Chen, Yadong Liu, Zongtan Zhou, Dewen Hu, and Ming Zhang. 2020. GAMA: Graph Attention Multi-agent Reinforcement Learning Algorithm for Cooperation. *Applied Intelligence* 50, 12 (July 2020), 4195–4205. doi:10.1007/s10489-020-01755-8

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

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

[18] Cédric Colas, Tristan Karch, Olivier Sigaud, and Pierre-Yves Oudeyer. 2022. Autotelic Agents with Intrinsically Motivated Goal-Conditioned Reinforcement Learning: A Short Survey. *Journal of Artificial Intelligence Research* 74 (July 2022), 1159–1199. doi:10.1613/jair.1.13554

[19] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. 2019. TarMAC: Targeted Multi-Agent Communication. In *Proc. of the 36th International Conference on Machine Learning*. PMLR, 1538–1546. https://proceedings.mlr.press/v97/das19a.html

[20] Peter Dayan and Geoffrey E Hinton. 1992. Feudal Reinforcement Learning. In *Advances in Neural Information Processing Systems*, Vol. 5. Morgan-Kaufmann. https://papers.nips.cc/paper/1992/hash/d14220ee66aeec73c49038385428ec4c-Abstract.html

[21] T. G. Dietterich. 2000. Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition. *Journal of Artificial Intelligence Research* 13 (Nov. 2000), 227–303. doi:10.1613/jair.639

[22] Wei Du and Shifei Ding. 2021. A Survey on Multi-Agent Deep Reinforcement Learning: From the Perspective of Challenges and Applications. *Artificial Intelligence Review* 54, 5 (June 2021), 3215–3238. doi:10.1007/s10462-020-09938-y

[23] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. 2021. First Return, Then Explore. *Nature* 590, 7847 (Feb. 2021), 580–586. doi:10.1038/s41586-020-03157-9

[24] Vladimir Egorov and Alexei Shpilman. 2022. Scalable Multi-Agent Model-Based Reinforcement Learning. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems* (Auckland, New Zealand) *(AAMAS '22)*. IFAAMAS, Richland, SC, USA, 381–390. https://dl.acm.org/doi/10.5555/3535850.3535894

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

[26] Chelsea Finn, P. Abbeel, and S. Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML 2017*.

[27] Chelsea Finn and Sergey Levine. 2018. Meta-Learning and Universality: Deep Representations and Gradient Descent Can Approximate Any Learning Algorithm. In *ICLR 2018 Poster*.

[28] Jakob Foerster, Nantas Nardelli, Gregory Farquhar, Triantafyllos Afouras, Philip H. S. Torr, Pushmeet Kohli, and Shimon Whiteson. 2017. Stabilising Experience Replay for Deep Multi-Agent Reinforcement Learning. In *Proc. of the 34th International Conference on Machine Learning - Volume 70 (ICML'17)*. JMLR.org, Sydney, NSW, Australia, 1146–1155.

[29] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. 2016. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. *arXiv:1605.06676 [cs]* (May 2016). arXiv:1605.06676 [cs]

[30] Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual Multi-Agent Policy Gradients. In *Proc. of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence (AAAI'18/IAAI'18/EAAI'18)*. AAAI Press, New Orleans, Louisiana, USA, 2974–2982.

[31] Matteo Gallici, Mario Martin, and Ivan Masmitja. 2023. TransfQMix: Transformers for Leveraging the Graph Structure of Multi-Agent Reinforcement Learning Problems. In *Proc. of the 2023 International Conference on Autonomous Agents and Multiagent Systems* (London,

United Kingdom) *(AAMAS '23)*. IFAAMAS, Richland, SC, USA, 1679–1687. https://dl.acm.org/doi/10.5555/3545946.3598825

[32] Sriram Ganapathi Subramanian, Pascal Poupart, Matthew E. Taylor, and Nidhi Hegde. 2020. Multi Type Mean Field Reinforcement Learning. In *Proc. of the 19th International Conference on Autonomous Agents and MultiAgent Systems* (Auckland, New Zealand) *(AAMAS '20)*. IFAAMAS, Richland, SC, USA, 411–419. https://dl.acm.org/doi/10.5555/3398761.3398813

[33] Minghong Geng, Shubham Pateria, Budhitama Subagdja, and Ah-Hwee Tan. 2024. HiSOMA: A Hierarchical Multi-Agent Model Integrating Self-Organizing Neural Networks with Multi-Agent Deep Reinforcement Learning. *Expert Systems with Applications* 252 (Oct. 2024), 124117. doi:10.1016/j.eswa.2024.124117

[34] Sven Gronauer and Klaus Diepold. 2021. Multi-Agent Deep Reinforcement Learning: A Survey. *Artificial Intelligence Review* (April 2021). doi:10.1007/s10462-021-09996-w

[35] Shanzhi Gu, Mingyang Geng, and Long Lan. 2021. Attention-Based Fault-Tolerant Approach for Multi-Agent Reinforcement Learning Systems. *Entropy* 23, 9 (Aug. 2021), 1133. doi:10.3390/e23091133

[36] Zhaohan Guo, Shantanu Thakoor, Miruna Pislar, Bernardo Avila Pires, Florent Altché, Corentin Tallec, Alaa Saade, Daniele Calandriello, Jean-Bastien Grill, Yunhao Tang, Michal Valko, Remi Munos, Mohammad Gheshlaghi Azar, and Bilal Piot. 2022. BYOL-Explore: Exploration by Bootstrapped Prediction. *Advances in Neural Information Processing Systems* 35 (Dec. 2022), 31855–31870.

[37] Jayesh K. Gupta, Maxim Egorov, and Mykel Kochenderfer. 2017. Cooperative Multi-agent Control Using Deep Reinforcement Learning. In *Autonomous Agents and Multiagent Systems*, Gita Sukthankar and Juan A. Rodriguez-Aguilar (Eds.). Vol. 10642. Springer International Publishing, Cham, 66–83. doi:10.1007/978-3-319-71682-4_5

[38] Nikunj Gupta, G. Srinivasaraghavan, Swarup Kumar Mohalik, Nishant Kumar, and Matthew E. Taylor. 2022. HAMMER: Multi-Level Coordination of Reinforcement Learning Agents via Learned Messaging. arXiv:2102.00824 [cs] doi:10.48550/arXiv.2102.00824

[39] Nico Gürtler, Dieter Büchler, and Georg Martius. 2021. Hierarchical Reinforcement Learning with Timed Subgoals. In *Advances in Neural Information Processing Systems*, Vol. 34. Curran Associates, Inc., 21732–21743.

[40] Joey Hejna, Pieter Abbeel, and Lerrel Pinto. 2023. Improving Long-Horizon Imitation through Instruction Prediction. *Proc. of the AAAI Conference on Artificial Intelligence* 37, 7 (June 2023), 7857–7865. doi:10.1609/aaai.v37i7.25951

[41] Pablo Hernandez-Leal, Michael Kaisers, Tim Baarslag, and Enrique Munoz de Cote. 2019. A Survey of Learning in Multiagent Environments: Dealing with Non-Stationarity. *arXiv preprint arXiv:1707.09183v2 [cs.MA]* (March 2019). arXiv:1707.09183v2 [cs.MA]

[42] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E. Taylor. 2019. A Survey and Critique of Multiagent Deep Reinforcement Learning. *Autonomous Agents and Multi-Agent Systems* 33, 6 (Nov. 2019), 750–797. doi:10.1007/s10458-019-09421-1

[43] Alexander Herzog, Kanishka Rao, Karol Hausman, Yao Lu, Paul Wohlhart, Mengyuan Yan, Jessica Lin, Montserrat Gonzalez Arenas, Ted Xiao, Daniel Kappler, Daniel Ho, Jarek Rettinghouse, Yevgen Chebotar, Kuang-Huei Lee, Keerthana Gopalakrishnan, Ryan Julian, Adrian Li, Chuyuan Kelly Fu, Bob Wei, Sangeetha Ramesh, Khem Holden, Kim Kleiven, David Rendleman, Sean Kirmani, Jeff Bingham, Jon Weisz, Ying Xu, Wenlong Lu, Matthew Bennice, Cody Fong, David Do, Jessica Lam, Yunfei Bai, Benjie Holson, Michael Quinlan, Noah Brown, Mrinal Kalakrishnan, Julian Ibarz, Peter Pastor, and Sergey Levine. 2023. Deep RL at Scale: Sorting Waste in Office Buildings with a Fleet of Mobile Manipulators. arXiv:2305.03270 [cs] doi:10.48550/arXiv.2305.03270

[44] Kai Hu, Keer Xu, Qingfeng Xia, Mingyang Li, Zhiqiang Song, Lipeng Song, and Ning Sun. 2024. An Overview: Attention Mechanisms in Multi-Agent Reinforcement Learning. *Neurocomputing* 598 (Sept. 2024), 128015. doi:10.1016/j.neucom.2024.128015

[45] Wenhan Huang, Kai Li, Kun Shao, Tianze Zhou, Matthew Taylor, Jun Luo, Dongge Wang, Hangyu Mao, Jianye Hao, Jun Wang, and Xiaotie Deng. 2022. Multiagent Q-learning with Sub-Team Coordination. In *Advances in Neural Information Processing Systems*, Vol. 35. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2022/hash/bd31bfd4caa85bffe07a35568182cdfa-Abstract-Conference.html

[46] Shariq Iqbal, Robby Costales, and Fei Sha. 2022. ALMA: Hierarchical Learning for Composite Multi-Agent Tasks. *Advances in Neural Information Processing Systems* 35 (Dec. 2022), 7155–7166.

[47] Shariq Iqbal and Fei Sha. 2019. Actor-Attention-Critic for Multi-Agent Reinforcement Learning. In *Proc. of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 2961–2970. https://proceedings.mlr.press/v97/iqbal19a.html

[48] Jeewon Jeon, Woojun Kim, Whiyoung Jung, and Youngchul Sung. 2022. MASER: Multi-Agent Reinforcement Learning with Subgoals Generated from Experience Replay Buffer. In *Proc. of the 39th International Conference on Machine Learning*. PMLR, 10041–10052.

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

[50] Xiangyu Kong, Bo Xin, Fangchen Liu, and Yizhou Wang. 2017. Revisiting the Master-Slave Architecture in Multi-Agent Deep Reinforcement Learning. *arXiv preprint arXiv:1712.07305v1 [cs.AI]* (Dec. 2017). arXiv:1712.07305v1 [cs.AI] doi:10.48550/arXiv.1712.07305

[51] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. 2016. Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. In *Advances in Neural Information Processing Systems*, Vol. 29. Curran Associates, Inc.

[52] Saurabh Kumar, Pararth Shah, Dilek Hakkani-Tur, and Larry Heck. 2017. Federated Control with Hierarchical Multi-Agent Deep Reinforcement Learning. *arXiv preprint arXiv:1712.08266v1 [cs.AI]* (Dec. 2017). arXiv:1712.08266v1 [cs.AI] doi:10.48550/arXiv.1712.08266

[53] Lior Kuyer, Shimon Whiteson, Bram Bakker, and Nikos Vlassis. 2008. Multiagent Reinforcement Learning for Urban Traffic Control Using Coordination Graphs. In *Machine Learning and Knowledge Discovery in Databases (Lecture Notes in Computer Science, Vol. 5211)*, Walter Daelemans, Bart Goethals, and Katharina Morik (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 656–671. doi:10.1007/978-3-540-87479-9_61

[54] Joel Z. Leibo, V. Zambaldi, Marc Lanctot, J. Marecki, and T. Graepel. 2017. Multi-Agent Reinforcement Learning in Sequential Social Dilemmas. In *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017)*. Sao Paulo, Brazil.

[55] Wenhao Li, Bo Jin, and Xiangfeng Wang. 2019. SparseMAAC: Sparse Attention for Multi-agent Reinforcement Learning. In *Database Systems for Advanced Applications (DASFAA 2019) (Lecture Notes in Computer Science, Vol. 11448)*. Springer, Cham, 96–110. doi:10.1007/978-3-030-18590-9_7

[56] Wenhao Li, Dan Qiao, Baoxiang Wang, Xiangfeng Wang, Bo Jin, and Hongyuan Zha. 2023. Semantically Aligned Task Decomposition in Multi-Agent Reinforcement Learning. arXiv:2305.10865 [cs] doi:10.48550/arXiv.2305.10865

[57] Wenhao Li, Xiangfeng Wang, Bo Jin, Junjie Sheng, Yun Hua, and Hongyuan Zha. 2021. Structured Diversification Emergence via Reinforced Organization Control and Hierachical Consensus Learning. In *Proc. of the 20th International Conference on Autonomous Agents and MultiAgent Systems* (London, United Kingdom) *(AAMAS '21)*. IFAAMAS, Richland, SC, USA, 773–781. https://dl.acm.org/doi/10.5555/3463952.3464045

[58] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous Control with Deep Reinforcement Learning. In *Conference Track Proceedings of the 4th International Conference on Learning Representations (ICLR 2016)* (San Juan, Puerto Rico). doi:10.48550/arXiv.1509.02971

[59] Michael L. Littman. 1994. Markov Games as a Framework for Multi-Agent Reinforcement Learning. In *Machine Learning Proceedings 1994*, William W. Cohen and Haym Hirsh (Eds.). Morgan Kaufmann, San Francisco (CA), 157–163. doi:10.1016/B978-1-55860-335-6.50027-1

[60] Bo Liu, Qiang Liu, Peter Stone, Animesh Garg, Yuke Zhu, and Anima Anandkumar. 2021. Coach-Player Multi-agent Reinforcement Learning for Dynamic Team Composition. In *Proc. of the 38th International Conference on Machine Learning (ICML 2021)* (Virtual Event) *(Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 6860–6870. https://proceedings.mlr.press/v139/liu21m.html

[61] Boyin Liu, Zhiqiang Pu, Yi Pan, Jianqiang Yi, Yanyan Liang, and D. Zhang. 2023. Lazy Agents: A New Perspective on Solving Sparse Reward Problem in Multi-agent Reinforcement Learning. In *Proc. of the 40th International Conference on Machine Learning*. PMLR, 21937–21950.

[62] Iou-Jen Liu, Unnat Jain, Raymond A. Yeh, and Alexander Schwing. 2021. Cooperative Exploration for Multi-Agent Deep Reinforcement Learning. In *Proc. of the 38th International Conference on Machine Learning*. PMLR, 6826–6836.

[63] Yilin Liu, Guiyang Luo, Quan Yuan, Jinglin Li, Lei Jin, Bo Chen, and Rui Pan. 2023. GPLight: Grouped Multi-agent Reinforcement Learning for Large-scale Traffic Signal Control. In *Proc. of the Thirty-Second International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, Macau, SAR China, 199–207. doi:10.24963/ijcai.2023/23

[64] Yong Liu, Weixun Wang, Yujing Hu, Jianye Hao, Xingguo Chen, and Yang Gao. 2020. Multi-Agent Game Abstraction via Graph Attention Neural Network. *Proc. of the AAAI Conference on Artificial Intelligence* 34, 05 (April 2020), 7211–7218. doi:10.1609/aaai.v34i05.6211

[65] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates Inc., 6379–6390. https://proceedings.neurips.cc/paper_files/paper/2017/file/68a9750337a418a86fe06c1991a1d64c-Paper.pdf

[66] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. 2019. MAVEN: Multi-Agent Variational Exploration. In *Advances in Neural Information Processing Systems* (Vancouver, Canada), Vol. 32. Curran Associates, Inc., Red Hook, NY, USA. https://proceedings.neurips.cc/paper/2019/hash/f816dc0acface7498e10496222e9db10-Abstract.html

[67] Rajbala Makar, Sridhar Mahadevan, and Mohammad Ghavamzadeh. 2001. Hierarchical Multi-Agent Reinforcement Learning. In *Proc. of the Fifth International Conference on Autonomous Agents* (Montreal, QC, Canada) *(AGENTS '01)*. ACM, New York, NY, USA, 246–253. doi:10.1145/375735.376302

[68] Aleksandra Malysheva, Daniel Kudenko, and Aleksei Shpilman. 2019. MAGNet: Multi-agent Graph Network for Deep Multi-agent Reinforcement Learning. In *2019 XVI International Symposium "Problems of Redundancy in Information and Control Systems" (REDUNDANCY)*. 171–176. doi:10.1109/REDUNDANCY48165.2019.9003345

[69] Hangyu Mao, Wulong Liu, Jianye Hao, Jun Luo, Dong Li, Zhengchao Zhang, Jun Wang, and Zhen Xiao. 2020. Neighborhood Cognition Consistent Multi-Agent Reinforcement Learning. *Proc. of the AAAI Conference on Artificial Intelligence* 34, 05 (June 2020), 7219–7226. doi:10.1609/aaai.v34i05.6212

[70] Hangyu Mao, Zhengchao Zhang, Zhen Xiao, and Zhibo Gong. 2019. Modelling the Dynamic Joint Policy of Teammates with Attention Multi-agent DDPG. In *Proc. of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (Montreal, QC, Canada) *(AAMAS '19)*. IFAAMAS, Richland, SC, USA, 1108–1116. https://dl.acm.org/doi/10.5555/3306127.3331810

[71] Hangyu Mao, Zhengchao Zhang, Zhen Xiao, Zhibo Gong, and Yan Ni. 2020. Learning Multi-Agent Communication with Double Attentional Deep Reinforcement Learning. *Autonomous Agents and Multi-Agent Systems* 34, 1 (March 2020), 1–34. doi:10.1007/s10458-

020-09455-w

[72] Hangyu Mao, Rui Zhao, Ziyue Li, Zhiwei Xu, Hao Chen, Yiqun Chen, Bin Zhang, Zhen Xiao, Junge Zhang, and Jiangjin Yin. 2024. PDiT: Interleaving Perception and Decision-making Transformers for Deep Reinforcement Learning. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems* (Auckland, New Zealand) *(AAMAS '24)*. IFAAMAS, Richland, SC, USA, 1363–1371. https://dl.acm.org/doi/10.5555/3635637.3662995

[73] Weichao Mao, Haoran Qiu, Chen Wang, Hubertus Franke, Zbigniew Kalbarczyk, Ravi Iyer, and Tamer Basar. 2023. Multi-Agent Meta-Reinforcement Learning: Sharper Convergence Rates with Task Similarity. In *NeurIPS 2023 Poster*.

[74] Laetitia Matignon, Guillaume J. Laurent, and Nadine Le Fort-Piat. 2012. Independent Reinforcement Learners in Cooperative Markov Games: A Survey Regarding Coordination Problems. *The Knowledge Engineering Review* 27, 1 (Feb. 2012), 1–31. doi:10.1017/S0269888912000057

[75] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. In *Proc. of the 33rd International Conference on Machine Learning (ICML 2016)* (New York, New York, USA) *(Proceedings of Machine Learning Research, Vol. 48)*. PMLR, 1928–1937. https://proceedings.mlr.press/v48/mniha16.html

[76] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. *arXiv preprint* (Dec. 2013). arXiv:1312.5602v1 [cs.LG] doi:10.48550/arXiv.1312.5602

[77] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-Level Control through Deep Reinforcement Learning. *Nature* 518, 7540 (Feb. 2015), 529–533. doi:10.1038/nature14236

[78] Igor Mordatch and Pieter Abbeel. 2018. Emergence of Grounded Compositional Language in Multi-Agent Populations. *Proc. of the AAAI Conference on Artificial Intelligence* 32, 1 (April 2018). doi:10.1609/aaai.v32i1.11492

[79] Ofir Nachum, Haoran Tang, Xingyu Lu, Shixiang Gu, Honglak Lee, and Sergey Levine. 2019. Why Does Hierarchy (Sometimes) Work So Well in Reinforcement Learning? *arXiv:1909.10618 [cs, stat]* (Dec. 2019). arXiv:1909.10618 [cs, stat]

[80] Dung Nguyen, Phuoc Nguyen, Svetha Venkatesh, and Truyen Tran. 2022. Learning to Transfer Role Assignment Across Team Sizes. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems* (Auckland, New Zealand) *(AAMAS '22)*. IFAAMAS, Richland, SC, USA, 963–971. https://dl.acm.org/doi/10.5555/3535850.3535958

[81] Thanh Thi Nguyen, Ngoc Duy Nguyen, and Saeid Nahavandi. 2020. Deep Reinforcement Learning for Multi-agent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Transactions on Cybernetics* 50, 9 (Sept. 2020), 3826–3839. doi:10.1109/TCYB.2020.2977374

[82] Yaru Niu, Rohan Paleja, and Matthew Gombolay. 2021. Multi-Agent Graph-Attention Communication and Teaming. In *Proc. of the 20th International Conference on Autonomous Agents and MultiAgent Systems* (London, United Kingdom) *(AAMAS '21)*. IFAAMAS, Richland, SC, USA, 964–973. https://dl.acm.org/doi/10.5555/3463952.3464065

[83] OpenAI. 2022. Introducing ChatGPT. https://openai.com/blog/chatgpt.

[84] Afshin Oroojlooy and Davood Hajinezhad. 2023. A Review of Cooperative Multi-Agent Deep Reinforcement Learning. *Applied Intelligence* 53, 11 (June 2023), 13677–13722. doi:10.1007/s10489-022-04105-y

[85] Gregory Palmer, Karl Tuyls, Daan Bloembergen, and Rahul Savani. 2018. Lenient Multi-Agent Deep Reinforcement Learning. In *Proc. of the 17th International Conference on Autonomous Agents and MultiAgent Systems* (Stockholm, Sweden) *(AAMAS '18)*. IFAAMAS, Richland, SC, USA, 443–451. https://dl.acm.org/doi/10.5555/3237383.3237451

[86] Georgios Papoudakis, Filippos Christianos, Arrasy Rahman, and Stefano V. Albrecht. 2019. Dealing with Non-Stationarity in Multi-Agent Deep Reinforcement Learning. *arXiv preprint* (June 2019). arXiv:1906.04737v1 [cs.LG] doi:10.48550/arXiv.1906.04737

[87] Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. 2021. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative Tasks. In *Proc. of the Neural Information Processing Systems Track on Datasets and Benchmarks*, Vol. 1. Curran Associates Inc.

[88] Fabio Pardo, Arash Tavakoli, Vitaly Levdik, and Petar Kormushev. 2018. Time Limits in Reinforcement Learning. In *Proc. of the 35th International Conference on Machine Learning*. PMLR, 4045–4054.

[89] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In *Proc. of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*. ACM, New York, NY, USA, 1–22. doi:10.1145/3586183.3606763

[90] Shubham Pateria, Budhitama Subagdja, Ah-hwee Tan, and Chai Quek. 2021. Hierarchical Reinforcement Learning: A Comprehensive Survey. *Comput. Surveys* 54, 5 (June 2021), 1–35. doi:10.1145/3453160

[91] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. 2017. Curiosity-Driven Exploration by Self-supervised Prediction. *arXiv:1705.05363 [cs, stat]* (May 2017). arXiv:1705.05363 [cs, stat]

[92] Peng Peng, Ying Wen, Yaodong Yang, Quan Yuan, Zhenkun Tang, Haitao Long, and Jun Wang. 2017. Multiagent Bidirectionally-Coordinated Nets: Emergence of Human-level Coordination in Learning to Play StarCraft Combat Games. *arXiv:1703.10069 [cs]* (Sept. 2017). arXiv:1703.10069 [cs]

[93] Leonid Peshkin, Kee-Eung Kim, Nicolas Meuleau, and Leslie Pack Kaelbling. 2000. Learning to Cooperate via Policy Search. In *Proc. of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI'00)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 489–496.

[94] Thomy Phan, Fabian Ritz, Lenz Belzner, Philipp Altmann, Thomas Gabor, and Claudia Linnhoff-Popien. 2021. VAST: Value Function Factorization with Variable Agent Sub-Teams. In *Advances in Neural Information Processing Systems*, Vol. 34. Curran Associates, Inc., 24018–24032.

[95] Aske Plaat. 2022. *Deep Reinforcement Learning* (1 ed.). Springer Nature, Singapore. https://link.springer.com/book/10.1007/978-981-19-0638-1

[96] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *Proc. of the 35th International Conference on Machine Learning (ICML 2018)* (Stockholm, Sweden) *(Proceedings of Machine Learning Research, Vol. 80)*. PMLR, 4295–4304. https://proceedings.mlr.press/v80/rashid18a.html

[97] Jingqing Ruan, Xiaotian Hao, Dong Li, and Hangyu Mao. 2023. Learning to Collaborate by Grouping: A Consensus-Oriented Strategy for Multi-Agent Reinforcement Learning. In *ECAI 2023* (Kraków, Poland), Vol. 372. 2010–2017. doi:10.3233/FAIA230493

[98] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. 2019. The StarCraft Multi-Agent Challenge. In *Proc. of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (Montreal, QC, Canada) *(AAMAS '19)*. IFAAMAS, Richland, SC, USA, 2186–2188. https://dl.acm.org/doi/10.5555/3306127.3332052

[99] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2016. Prioritized Experience Replay. In *Conference Track Proceedings of the 4th International Conference on Learning Representations (ICLR 2016)* (San Juan, Puerto Rico). doi:10.48550/arXiv.1511.05952

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

[101] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust Region Policy Optimization. In *Proc. of the 32nd International Conference on Machine Learning (ICML 2015)* (Lille, France), Vol. 37. PMLR, 1889–1897. https://proceedings.mlr.press/v37/schulman15.html

[102] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv:1707.06347 [cs]* (Aug. 2017). arXiv:1707.06347 [cs]

[103] Jianzhun Shao, Zhiqiang Lou, Hongchang Zhang, Yuhang Jiang, Shuncheng He, and Xiangyang Ji. 2022. Self-Organized Group for Cooperative Multi-agent Reinforcement Learning. In *Advances in Neural Information Processing Systems 35*, Vol. 35. Curran Associates, Inc., New Orleans, Louisiana, USA, 5711–5723.

[104] Yoav Shoham and Kevin Leyton-Brown. 2008. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, Cambridge.

[105] Felipe Leno Da Silva and Anna Helena Reali Costa. 2019. A Survey on Transfer Learning for Multiagent Reinforcement Learning Systems. *Journal of Artificial Intelligence Research* 64 (March 2019), 645–703. doi:10.1613/jair.1.11396

[106] Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. 2019. Learning When to Communicate at Scale in Multiagent Cooperative and Competitive Tasks. In *Conference Track Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)* (New Orleans, Louisiana, USA). https://openreview.net/forum?id=rye7knCqK7

[107] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Hostallero, and Yung Yi. 2019. QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning. In *Proc. of the 36th International Conference on Machine Learning (ICML 2019)* (Long Beach, California, USA) *(Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 5887–5896. http://proceedings.mlr.press/v97/son19a.html

[108] Joseph Suarez, Yilun Du, Phillip Isola, and Igor Mordatch. 2019. Neural MMO: A Massively Multiagent Game Environment for Training and Evaluating Intelligent Agents. arXiv:1903.00784 [cs, stat] doi:10.48550/arXiv.1903.00784

[109] Sainbayar Sukhbaatar, arthur szlam, and Rob Fergus. 2016. Learning Multiagent Communication with Backpropagation. In *Advances in Neural Information Processing Systems*, Vol. 29. Curran Associates, Inc.

[110] Chuangchuang Sun, Macheng Shen, and Jonathan P. How. 2020. Scaling Up Multiagent Reinforcement Learning for Robotic Systems: Learn an Adaptive Sparse Communication Graph. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 11755–11762. doi:10.1109/IROS45743.2020.9341303

[111] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. 2018. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *Proc. of the 17th International Conference on Autonomous Agents and MultiAgent Systems* (Stockholm, Sweden) *(AAMAS '18)*. IFAAMAS, Richland, SC, USA, 2085–2087. https://dl.acm.org/doi/10.5555/3237383.3238080

[112] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (second edition ed.). The MIT Press, Cambridge, Massachusetts.

[113] Gabriel Synnaeve, Nantas Nardelli, Alex Auvolat, Soumith Chintala, Timothée Lacroix, Zeming Lin, Florian Richoux, and Nicolas Usunier. 2016. TorchCraft: A Library for Machine Learning Research on Real-Time Strategy Games. arXiv:1611.00625 [cs] doi:10.48550/arXiv.1611.00625

[114] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. 2017. Multiagent Cooperation and Competition with Deep Reinforcement Learning. *PLOS ONE* 12, 4 (April 2017), 1–15. doi:10.1371/journal.pone.0172395

[115] Ming Tan. 1993. Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. In *Machine Learning Proceedings 1993*. Morgan Kaufmann, San Francisco (CA), 330–337. doi:10.1016/B978-1-55860-307-3.50049-6

[116] Hongyao Tang, Jianye Hao, Tangjie Lv, Yingfeng Chen, Zongzhang Zhang, Hangtian Jia, Chunxu Ren, Yan Zheng, Zhaopeng Meng, Changjie Fan, and Li Wang. 2019. Hierarchical Deep Multiagent Reinforcement Learning with Temporal Abstraction. arXiv:1809.09332 [cs] doi:10.48550/arXiv.1809.09332

[117] Nicolas Usunier, Gabriel Synnaeve, Zeming Lin, and Soumith Chintala. 2016. Episodic Exploration for Deep Deterministic Policies: An Application to StarCraft Micromanagement Tasks. In *ICLR 2017 Poster*. arXiv:1609.02993

[118] Elise Van der Pol and Frans A Oliehoek. 2016. Coordinated Deep Reinforcement Learners for Traffic Light Control. In *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)*, Vol. 8. 21–38.

[119] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems* (Long Beach, California, USA), Vol. 30. Curran Associates, Inc., Red Hook, NY, USA. https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[120] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *Conference Track Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)* (Vancouver, Canada). https://openreview.net/forum?id=rJXMpikCZ

[121] Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. 2017. FeUdal Networks for Hierarchical Reinforcement Learning. In *Proc. of the 34th International Conference on Machine Learning*. PMLR, 3540–3549.

[122] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy Lillicrap, Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, and Rodney Tsing. 2017. StarCraft II: A New Challenge for Reinforcement Learning. arXiv:1708.04782v1 [cs.LG]

[123] Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. 2021. QPLEX: Duplex Dueling Multi-Agent Q-Learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Rcmk0xxIQV

[124] Juzhen Wang, Xiaoli Zhang, Xingshi He, and Yongqiang Sun. 2023. Bandwidth Allocation and Trajectory Control in UAV-Assisted IoV Edge Computing Using Multiagent Reinforcement Learning. *IEEE Transactions on Reliability* 72, 2 (June 2023), 599–608. doi:10.1109/TR.2022.3192020

[125] Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. 2020. ROMA: Multi-Agent Reinforcement Learning with Emergent Roles. In *Proc. of the 37th International Conference on Machine Learning (ICML 2020)* (Virtual Event) *(Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 9876–9886. https://proceedings.mlr.press/v119/wang20f.html

[126] Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. 2021. RODE: Learning Roles to Decompose Multi-Agent Tasks. In *International Conference on Learning Representations* (Virtual Event). https://openreview.net/forum?id=TTUVg6vkNjK

[127] Christopher John Cornish Hellaby Watkins. 1989. *Learning from Delayed Rewards*. Ph. D. Dissertation.

[128] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, Vol. 35. Curran Associates, Inc., 24824–24837. https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf

[129] Gerhard Weiß. 1995. Distributed Reinforcement Learning. In *The Biology and Technology of Intelligent Autonomous Agents (NATO ASI Series)*, Luc Steels (Ed.). Springer, Berlin, Heidelberg, 415–428. doi:10.1007/978-3-642-79629-6_18

[130] Muning Wen, Jakub Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. 2022. Multi-Agent Reinforcement Learning Is a Sequence Modeling Problem. In *Advances in Neural Information Processing Systems*, Vol. 35. Curran Associates, Inc., 16509–16521. https://proceedings.neurips.cc/paper_files/paper/2022/hash/69413f87e5a34897cd010ca698097d0a-Abstract-Conference.html

[131] Annie Wong, Thomas Bäck, Anna V. Kononova, and Aske Plaat. 2023. Deep Multiagent Reinforcement Learning: Challenges and Directions. *Artificial Intelligence Review* 56, 6 (June 2023), 5023–5056. doi:10.1007/s10462-022-10299-x

[132] Cathy Wu, Aboudy Kreidieh, Kanaad Parvate, Eugene Vinitsky, and Alexandre M. Bayen. 2022. Flow: A Modular Learning Framework for Mixed Autonomy Traffic. *IEEE Transactions on Robotics* 38, 2 (April 2022), 1270–1286. arXiv:1710.05465 [cs] doi:10.1109/TRO.2021.3087314

[133] Tong Wu, Pan Zhou, Kai Liu, Yali Yuan, Xiumin Wang, Huawei Huang, and Dapeng Oliver Wu. 2020. Multi-Agent Deep Reinforcement Learning for Urban Traffic Light Control in Vehicular Networks. *IEEE Transactions on Vehicular Technology* 69, 8 (Aug. 2020), 8243–8256.

doi:10.1109/TVT.2020.2997896

[134] Jian Xiao, Guohui Yuan, Jinhui He, Kai Fang, and Zhuoran Wang. 2023. Graph Attention Mechanism Based Reinforcement Learning for Multi-Agent Flocking Control in Communication-Restricted Environment. *Information Sciences* 620 (Jan. 2023), 142–157. doi:10.1016/j.ins.2022.11.059

[135] Yuchen Xiao, Weihao Tan, and Christopher Amato. 2022. Asynchronous Actor-Critic for Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems*, Vol. 35. Curran Associates, Inc., New Orleans, Louisiana, USA, 4385–4400.

[136] Pei Xu, Junge Zhang, and Kaiqi Huang. 2023. Exploration via Joint Policy Diversity for Sparse-Reward Multi-Agent Tasks. In *Proc. of the Thirty-Second International Joint Conference on Artificial Intelligence Main Track*, Vol. 1. 326–334. doi:10.24963/ijcai.2023/37

[137] Pei Xu, Junge Zhang, Qiyue Yin, Chao Yu, Yaodong Yang, and Kaiqi Huang. 2023. Subspace-Aware Exploration for Sparse-Reward Multi-Agent Tasks. *Proc. of the AAAI Conference on Artificial Intelligence* 37, 10 (June 2023), 11717–11725. doi:10.1609/aaai.v37i10.26384

[138] Zhiwei Xu, Yunpeng Bai, Bin Zhang, Dapeng Li, and Guoliang Fan. 2023. HAVEN: Hierarchical Cooperative Multi-Agent Reinforcement Learning with Dual Coordination Mechanism. *Proc. of the AAAI Conference on Artificial Intelligence* 37, 10 (June 2023), 11735–11743. doi:10.1609/aaai.v37i10.26386

[139] Zhiwei Xu, Dapeng Li, Bin Zhang, Yuan Zhan, Yunpeng Baiia, and Guoliang Fan. 2022. Mingling Foresight with Imagination: Model-Based Cooperative Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems 35*, Vol. 15. Curran Associates, Inc., New Orleans, Louisiana, USA, 11327–11340.

[140] Zhao Xu, Yang Lyu, Quan Pan, Jinwen Hu, Chunhui Zhao, and Shuai Liu. 2018. Multi-Vehicle Flocking Control with Deep Deterministic Policy Gradient Method. In *2018 IEEE 14th International Conference on Control and Automation (ICCA)*. 306–311. doi:10.1109/ICCA.2018.8444355

[141] Jiachen Yang, Igor Borovikov, and Hongyuan Zha. 2020. Hierarchical Cooperative Multi-Agent Reinforcement Learning with Skill Discovery. In *Proc. of the 19th International Conference on Autonomous Agents and MultiAgent Systems* (Auckland, New Zealand) *(AAMAS '20)*. IFAAMAS, Richland, SC, USA, 1566–1574. https://dl.acm.org/doi/10.5555/3398761.3398941

[142] Mingyu Yang, Jian Zhao, Xunhan Hu, Wengang Zhou, Jiangcheng Zhu, and Houqiang Li. 2022. LDSA: Learning Dynamic Subtask Assignment in Cooperative Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems 35*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., New Orleans, Louisiana, USA, 1698–1710.

[143] Shantian Yang and Bo Yang. 2022. An Inductive Heterogeneous Graph Attention-Based Multi-Agent Deep Graph Infomax Algorithm for Adaptive Traffic Signal Control. *Information Fusion* 88 (Dec. 2022), 249–262. doi:10.1016/j.inffus.2022.08.001

[144] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. 2018. Mean Field Multi-Agent Reinforcement Learning. In *Proc. of the 35th International Conference on Machine Learning (ICML 2018)* (Stockholm, Sweden) *(Proceedings of Machine Learning Research, Vol. 80)*. PMLR, 5571–5580. https://proceedings.mlr.press/v80/yang18d.html

[145] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In *Advances in Neural Information Processing Systems 35*, Vol. 32. Curran Associates, Inc., New Orleans, Louisiana, USA, 24611–24624.

[146] Won Joon Yun, Jihong Park, and Joongheon Kim. 2023. Quantum Multi-Agent Meta Reinforcement Learning. *Proc. of the AAAI Conference on Artificial Intelligence* 37, 9 (June 2023), 11087–11095. doi:10.1609/aaai.v37i9.26313

[147] Bin Zhang, Weihao Hu, Amer M.Y.M. Ghias, Xiao Xu, and Zhe Chen. 2023. Multi-Agent Deep Reinforcement Learning Based Distributed Control Architecture for Interconnected Multi-Energy Microgrid Energy Management and Optimization. *Energy Conversion and Management* 277 (Feb. 2023), 116647. doi:10.1016/j.enconman.2022.116647

[148] Bin Zhang, Hangyu Mao, Lijuan Li, Zhiwei Xu, Dapeng Li, Rui Zhao, and Guoliang Fan. 2024. Sequential Asynchronous Action Coordination in Multi-Agent Systems: A Stackelberg Decision Transformer Approach. In *Proc. of the 41st International Conference on Machine Learning (ICML 2024)* (Vienna, Austria) *(Proceedings of Machine Learning Research, Vol. 235)*. PMLR, 59559–59575. https://openreview.net/forum?id=M3qRRkOuTN

[149] Huichu Zhang, Siyuan Feng, Chang Liu, Yaoyao Ding, Yichen Zhu, Zihan Zhou, Weinan Zhang, Yong Yu, Haiming Jin, and Zhenhui Li. 2019-05-13/2019-05-17. CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario. In *The World Wide Web Conference (WWW '19)*. ACM, New York, NY, USA, 3620–3624. doi:10.1145/3308558.3314139

[150] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. 2021. Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. In *Handbook of Reinforcement Learning and Control*, Kyriakos G. Vamvoudakis, Yan Wan, Frank L. Lewis, and Derya Cansever (Eds.). Studies in Systems, Decision and Control, Vol. 325. Springer International Publishing, Cham, 321–384. doi:10.1007/978-3-030-60990-0_12

[151] Xianjie Zhang, Yu Liu, Xiujuan Xu, Qiong Huang, Hangyu Mao, and Anil Carie. 2021. Structural Relational Inference Actor-Critic for Multi-Agent Reinforcement Learning. *Neurocomputing* 459 (Oct. 2021), 383–394. doi:10.1016/j.neucom.2021.07.014

[152] Lulu Zheng, Jiarui Chen, Jianhao Wang, Jiamin He, Yujing Hu, Yingfeng Chen, Changjie Fan, Yang Gao, and Chongjie Zhang. 2021. Episodic Multi-agent Reinforcement Learning with Curiosity-Driven Exploration. arXiv:2111.11032 [cs] doi:10.48550/arXiv.2111.11032

[153] Yan Zheng, Zhaopeng Meng, Jianye Hao, and Zongzhang Zhang. 2018. Weighted Double Deep Multiagent Reinforcement Learning in Stochastic Cooperative Environments. In *PRICAI 2018: Trends in Artificial Intelligence (Lecture Notes in Computer Science)*, Xin Geng and Byeong-Ho Kang (Eds.). Springer International Publishing, Cham, 421–429. doi:10.1007/978-3-319-97310-4_48

[154] Ziyuan Zhou, Guanjun Liu, and Mengchu Zhou. 2024. A Robust Mean-Field Actor-Critic Reinforcement Learning Against Adversarial Perturbations on Agent States. *IEEE Transactions on Neural Networks and Learning Systems* 35, 10 (Oct. 2024), 14370–14381. doi:10.1109/TNNLS.2023.3278715

Received 21 February 2024; revised 5 May 2025; accepted 29 March 2026