Autonomous Agents and Multi-Agent Systems (2024) 38:4
https://doi.org/10.1007/s10458-023-09633-6

A survey of multi-agent deep reinforcement learning with communication

Changxi Zhu[1] · **Mehdi Dastani**[1] · **Shihan Wang**[1]

Accepted: 7 December 2023

Abstract
Communication is an effective mechanism for coordinating the behaviors of multiple agents, broadening their views of the environment, and to support their collaborations. In the field of multi-agent deep reinforcement learning (MADRL), agents can improve the overall learning performance and achieve their objectives by communication. Agents can communicate various types of messages, either to all agents or to specific agent groups, or conditioned on specific constraints. With the growing body of research work in MADRL with communication (Comm-MADRL), there is a lack of a systematic and structural approach to distinguish and classify existing Comm-MADRL approaches. In this paper, we survey recent works in the Comm-MADRL field and consider various aspects of communication that can play a role in designing and developing multi-agent reinforcement learning systems. With these aspects in mind, we propose 9 dimensions along which Comm-MADRL approaches can be analyzed, developed, and compared. By projecting existing works into the multi-dimensional space, we discover interesting trends. We also propose some novel directions for designing future Comm-MADRL systems through exploring possible combinations of the dimensions.

Keywords Multi-agent reinforcement learning · Deep reinforcement learning · Communication · Survey

1 Introduction

Many real-world scenarios, such as autonomous driving [1], sensor networks [2], robotics [3] and game-playing [4, 5], can be modeled as multi-agent systems. Such multi-agent systems can be designed and developed using multi-agent reinforcement learning (MARL) techniques to learn the behavior of individual agents, which can be cooperative,

✉ Changxi Zhu
c.zhu@uu.nl

✉ Mehdi Dastani
m.m.dastani@uu.nl

✉ Shihan Wang
s.wang2@uu.nl

[1] Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands

Published online: 06 January 2024

competitive, or a mixture of them. As agents are often distributed in the environment where they only have access to their local observations rather than the complete state of the environment, partial observability becomes an essential assumption in MARL [6–8]. Moreover, MARL suffers from the non-stationary issue [9], since each agent faces a dynamic environment that can be influenced by the changing and adapting policies of other agents. Communication has been viewed as a vital means to tackle the problems of partial observability and non-stationary in MARL. Agents can communicate individual information, e.g., observations, intentions, experiences, or derived features, to have a broader view of the environment, which in turn allows them to make well-informed decisions [9, 10].

Due to the recent success of deep learning [11] and its application to reinforcement learning [12], multi-agent deep reinforcement learning (MADRL) has witnessed great achievements in recent years, where agents can process high-dimensional data and have generalization ability in large state and action spaces [7, 8]. We notice that a large number of research works focus on *learning tasks with communication*, which aim at learning to solve domain-specific tasks, such as navigation, traffic, and video games, by communicating and sharing information. To the best of our knowledge, there is a lack of survey literature that can cover recent works on learning tasks with communication in multi-agent deep reinforcement learning (Comm-MADRL). Early surveys consider the role of communication in MARL but assume it to be predefined rather than a subject of learning [13–15]. Most Comm-MADRL surveys cover only a small number of research works without proposing a fine-grained classification system to compare and analyze them.[1] In cooperative scenarios, Hernandez-Leal et al. [16] use *learning communication* to denote the area of learning communication protocols to promote the cooperation of agents.[2] The only survey that we found classifying some early works in Comm-MADRL is from Gronauer and Diepold [17], which is based on distinguishing whether messages are received by all agents, a set of agents, or a network of agents. However, other aspects of Comm-MADRL, such as the type of messages and training paradigms, which are essential for communication and can help characterize existing communication protocols, are ignored. As a result, the reviewed papers in recent surveys regarding learning tasks with communication are rather limited and the proposed categorizations are too narrow to distinguish existing works in Comm-MADRL. On the other hand, there is a closely related research area, *emergent language/communication*, which also considers learning communication through various reinforcement learning techniques [18]. Different from Comm-MADRL, the primary goal of emergent language studies is to learn a symbolic language.[3] However, a subset of emergent language research works pursues an additional goal to leverage learnable symbolic language to enhance task-level performance. Notably, these research works have not been encompassed within existing Comm-MADRL surveys but included in our survey, referred to *learning tasks with emergent language*. In summary, our survey overlaps in scope with surveys of emergent language (i.e., in learning tasks with emergent language), but our survey focuses on different primary goals (i.e., achieving domain-specific tasks rather than

[1] We provide a detailed comparison of recent surveys on MADRL which involves communication in Sect. 2.3.

[2] In our survey, we extend the concept of *learning communication* to general multi-agent tasks and use the term *learning tasks with communication* to emphasize that the primary goal of recent research, which is centered on solving specific domain tasks through the use of communication.

[3] In the literature, *emergent language* and *emergent communication* are used interchangeably. In our survey, we use *emergent language* for referring to both terms.

learning a symbolic language). We further clarify the differences between learning tasks with communication and emergent language in Sect. 2.2.

In our survey paper, we review the Comm-MADRL literature by focusing on how communication can be utilized to improve the performance of multi-agent deep reinforcement learning techniques. Specifically, we focus on learnable communication protocols, which are aligned with recent works that emphasize the development of dynamic and adaptive communication, including learning when, how, and what to communicate with deep reinforcement learning techniques. Through a comprehensive review of recent Comm-MADRL literature, we propose a systematic and structured classification methodology designed to differentiate and categorize various Comm-MADRL approaches. Such a methodology will also provide guidance for the design and advancement of new Comm-MADRL systems. Suppose we plan to develop a Comm-MADRL system for a domain task at hand. Starting with the questions of when, how, and what to communicate, the system can be characterized from various aspects. Agents need to learn when to communicate, with whom to communicate, what information to convey, how to integrate received information, and, lastly, what learning objectives can be achieved through communication. We propose 9 dimensions that correspond to unique aspects of Comm-MADRL systems: Controlled Goals, Communication Constraints, Communicatee Type, Communication Policy, Communicated Messages, Message Combination, Inner Integration, Learning Methods, and Training Schemes. These dimensions, which form the skeleton of a Comm-MADRL system, can be used to analyze and gain insights into designed Comm-MADRL approaches thoroughly. By mapping recent Comm-MADRL approaches into this multi-dimensional structure, we not only provide insight into the current state of the art in this field but also determine some important directions for designing future Comm-MADRL systems.

The remaining sections of this paper are organized as follows. In Sect. 2 the preliminaries of multi-agent RL are discussed, together with existing extensions regarding communication and a detailed comparison of recent surveys. In Sect. 3, we present our proposed dimensions, explaining how we group the recent works in the categories of each dimension. In Sect. 4, we discuss the trends that we found in the literature, and, driven by the proposed dimensions, we propose possible research directions in this research area. We finalize the paper with some conclusions in Sect. 5.

2 Background

In this section, we first provide the necessary background on multi-agent reinforcement learning. Then, we show how multi-agent reinforcement learning can be extended to consider communication between agents. Finally, we present and compare recent surveys involving communication, from which we can directly see our motivations to fill the gaps among existing surveys.

2.1 Multi-agent reinforcement learning

Real-world applications often contain more than one agent that operate in the environment. Agents are generally assumed to be autonomous and required to learn their strategies for achieving their goals. A multi-agent environment can be formalized in several ways [6], depending on whether the environment is fully observable, how agents' goals are correlated, etc. Among them, the Partially Observable Stochastic Game

(POSG) [19, 20] is one of the most flexible formalizations. A POSG is defined by a tuple $\langle \mathcal{I}, \mathcal{S}, \rho^0, \{\mathcal{A}_i\}, P, \{\mathcal{O}_i\}, O, \{R_i\} \rangle$, where $\mathcal{I}$ is a (finite) set of agents indexed as $\{1, ..., n\}$, $\mathcal{S}$ is a set of environment states, ρ^0 is the initial state distribution over state space $\mathcal{S}$, $\mathcal{A}_i$ is a set of actions available to agent i, and $\mathcal{O}_i$ is a set of observations of agent i. We denote a joint action space as $\boldsymbol{\mathcal{A}} = \times_{i \in \mathcal{I}} \mathcal{A}_i$ and a joint observation space of agents as $\boldsymbol{\mathcal{O}} = \times_{i \in \mathcal{I}} \mathcal{O}_i$. Therefore, $P : \mathcal{S} \times \boldsymbol{\mathcal{A}} \to \Delta(\mathcal{S})$ denotes the transition probability from a state $s \in \mathcal{S}$ to a new state $s' \in \mathcal{S}$ given agents' joint action $\vec{a} = \langle a_1, ..., a_n \rangle$, where $\vec{a} \in \boldsymbol{\mathcal{A}}$. With the environment transitioning to the new state s', the probability of observing a joint observation $\vec{o} = \langle o_1, ..., o_n \rangle$ (where $\vec{o} \in \boldsymbol{\mathcal{O}}$) given the joint action $\vec{a}$ is determined according to the observation probability function $O : \mathcal{S} \times \boldsymbol{\mathcal{A}} \to \Delta(\boldsymbol{\mathcal{O}})$. Each agent then receives an immediate reward according to their own reward functions $R_i : \mathcal{S} \times \boldsymbol{\mathcal{A}} \times \mathcal{S} \to \mathbb{R}$. Similar to the joint action and observation, we could denote $\vec{r} = \langle r_1, ..., r_n \rangle$ as a joint reward. If agents' reward functions happen to be the same, i.e., they have identical goals, then $r_1 = r_2 = ... = r_n$ holds for every time step. In this setting, the POSG is reduced to a Dec-POMDP [6]. If at every time step the state is uniquely determined from the current set of observations of agents, i.e., $s \equiv \vec{o}$, the Dec-POMDP is reduced to a Dec-MDP. If each agent knows what the true environment state is, the Dec-MDP is reduced to a Multi-agent MDP. If there is only one single agent in the set of agents, i.e., $\mathcal{I} = \{1\}$, then the Multi-agent MDP is reduced to an MDP and the Dec-POMDP is reduced to a POMDP. Due to the partial observability, MARL methods often use the observation-action history $\tau_{i,t} = \{o_{i,0}, a_{i,0}, o_{i,1}, ..., o_{i,t}\}$ up to time step t for each agent to approximate the environment state. Note that time step t is often omitted for the sake of simplification.

In the multi-agent reinforcement learning setting, agents can learn their policies in either a decentralized or a centralized fashion. In decentralized learning (e.g., decentralized Q-learning [21, 22]), an n-agent MARL problem is decomposed into n decentralized single-agent problems where each agent learns its own policy by considering all other agents as a part of the environment [23, 24]. In such a decentralized setting, the learned policy of each agent is conditioned on its local observation and history. A major problem with decentralized learning is the so-called non-stationarity of the environment, i.e., the fact that each agent learns in an environment where other agents are simultaneously exploring and learning. Centralized learning enables the training of either a single joint policy for all agents or a centralized value function to facilitate the learning of n decentralized policies. While centralized (joint) learning removes or mitigates issues of partial observability and non-stationarity, it faces the challenge of joint action (and observation) spaces that expand exponentially with the number of agents and their actions. For a deeper dive into various training schemes used in MARL, we recommend the comprehensive survey by [17], which offers valuable insights into the training and execution of policies. Based on whether policies are derived from value functions or directly learned, multi-agent reinforcement learning methods can be categorized into value-based and policy-based methods. Both methods have been largely utilized in Comm-MADRL.

Value-based

Value-based methods in the multi-agent case borrow considerable ideas from the single-agent case. As one of the most popular value-based algorithms, the decentralized Q-learning learns a local Q-function for each agent. In the cooperative setting where agents share a common reward, the update rule for agent i is as follows:

$$Q_i(s, a_i) \leftarrow Q_i(s, a_i) + \alpha(\underbrace{r + \gamma \max_{a_i'} Q_i(s', a_i')}_{\text{new estimate}} - \underbrace{Q_i(s, a_i)}_{\text{current estimate}}) \tag{1}$$

where r is the shared reward, and a_i' is the action with the highest Q-value in the next state s'. In partially observable environments, the environment state is not fully observable and is usually replaced by the individual observation or history of each agent. The Q-values for each state-action pair are incrementally updated according to the TD error. This error, i.e., $r + \gamma \max_{a_i'} Q_i(s', a_i') - Q_i(s, a_i)$, represents the difference between a new estimate (i.e., $r + \gamma \max_{a_i'} Q_i(s', a_i')$) and the current estimate (i.e., $Q_i(s, a_i)$) based on the Bellman equation [25]. As the state and action space could be too large to be encountered frequently for accurate estimation, function approximation methods, like deep neural networks, have become popular for endowing value or policy models with generalization abilities across both discrete and continuous states and actions [12]. For example, the Deep Q-network (DQN) [12] minimizes the difference between the new estimate calculated from sampled rewards and the current estimate of a parameterized Q-function. In DQN-based methods, the Q-function in Eq. 1 is notated as $Q_i(s, a_i; \theta_i)$, which depends on learnable parameters θ_i. On the other hand, centralized learning in value-based methods learns a joint Q-function $Q(s, \vec{a}; \theta)$ with parameters θ. However, this approach can be challenging to scale with an increasing number of agents. Value decomposition methods [26–29] are popular MARL methods that decompose a joint Q-function to enable efficient training. These methods are also widely employed in research works in Comm-MADRL [30–32]. In partially observable environments, linear value decomposition methods decompose history-based joint Q-functions as follows:

$$Q^{joint}(\vec{\tau}, \vec{a}) = \sum_{i}^{n} w_i Q^i(\tau_i, a_i) \tag{2}$$

where the joint Q-function is based on the joint history of all agents and is decomposed into local Q-functions based on individual histories. The weight w_i can either be a fixed value [26, 28] or a learnable parameter subject to certain constraints [29]. Advantage functions can also replace the Q-function in the above equation to reduce variance [33].

Policy-based

Policy-based methods directly search over the policy space instead of obtaining the policy through value functions implicitly. The policy gradient theorem [25] provides an analytical expression of the gradients for a stochastic policy with learnable parameters in single-agent cases. In the multi-agent case with centralized learning, the policy gradient theorem is expressed as follows:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\vec{a} \sim \pi(\cdot|s), s \sim \rho^\pi}[\nabla_\theta \log \pi(\vec{a} \mid s; \theta) Q^\pi(s, \vec{a})] \tag{3}$$

where $J(\theta)$ represents the learning objective, and $\pi(\vec{a} \mid s; \theta)$ denotes a stochastic policy parameterized by θ (abbreviated as π). Additionally, ρ^π signifies the state distribution under the policy π, and $\nabla_\theta J(\theta)$ represents the expected gradient with respect to all possible actions and states. Due to the computational intractability of the expected gradient, stochastic gradient ascent can be applied to update the parameters θ at every learning step l as follows:

$$\theta_{l+1} = \theta_l + \alpha \widehat{\nabla_\theta J(\theta)}$$

where α is the learning rate, and $\widehat{\nabla_\theta J(\theta)}$ is an estimate of the expected gradient based on sampled actions and states. Moreover, the Q-function in Eq. 3 can be replaced by average returns over episodes to form REINFORCE algorithms [25], or by an estimated value function to form actor-critic algorithms [34, 35]. In actor-critic methods, the policy and value function are termed the actor and the critic, respectively. The critic will, therefore, guide the learning of the actor.

Actor-critic methods have undergone various adaptations for multi-agent environments [7, 8, 36, 37]. A typical extension is the multi-agent deep deterministic policy gradient (MADDPG) [7]. In MADDPG, the critic is a centralized Q-function designed to capture global information and coordinate learning signals. Meanwhile, the actors are local policies, ensuring decentralized execution. MADDPG assumes deterministic actors with continuous actions, allowing for the backpropagation of gradients from the value function to the policies. The gradient of each parameterized actor $\mu_{\theta_i}(a_i \mid o_i)$ with learnable parameters θ_i, abbreviated as μ_i, is defined as follows:

$$\nabla_{\theta_i} J(\theta_i) = \mathbb{E}_{\vec{o},\vec{a}\sim\mathcal{D}}\Big[\nabla_{\theta_i}\mu_i(a_i \mid o_i)\nabla_{a_i} Q_i^\mu(\vec{o}, a_1, \ldots, a_N) \mid_{a_i=\mu_i(o_i)}\Big]$$

where $\mathcal{D}$ is the experience buffer that contains joint observation-action tuples $\langle \vec{o}, \vec{a}, \vec{r}, \vec{o'} \rangle$. Each agent's Q-function, denoted as $Q_i^\mu(\vec{o}, a_1, \ldots, a_N)$, takes joint observations and actions as inputs, while decentralized actors use local observations as inputs. Contrary to Eq. 3, gradients with respect to the current action of agent i (specifically, $\mu_i(o_i)$) are utilized to guide the update of the policy parameter θ_i. Both MADDPG and its single-agent counterpart, DDPG, have seen widespread application in Comm-MADRL [38–42].

2.2 Extensions with communication

In the MADRL literature where communication is used, we notice two closely related research areas, which we will refer to with the terms *emergent language* and *learning tasks with communication*. The *emergent language* research area [18, 43–46] aims at learning a language grounded on symbols in communities of interacting/communicating agents. This line of research tries to understand the evolution of the language in agents equipped with neural networks. On the other hand, learning tasks with communication [16, 47–49] focuses primarily on solving multi-agent reinforcement learning tasks with the aid of communication. Communication is often regarded as information exchange rather than learning a (human-like) language. Despite the distinction, when using MADRL techniques on specific domain tasks, languages might emerge, which can potentially enhance the learning system's explainability in accomplishing those tasks. We illustrate the research areas, emergent language and learning tasks with communication, along with their intersection *learning tasks with emergent language* in Fig. 1. Notably, our survey focuses on learning tasks with communication in multi-agent deep reinforcement learning, including the intersection with emergent language.[4] Within this focus, multiple agents often operate in partially observable environments and learn to share information encoded through neural networks.

[4] Throughout the remainder of our survey, Comm-MADRL will be used to specifically refer to the areas of our focus.

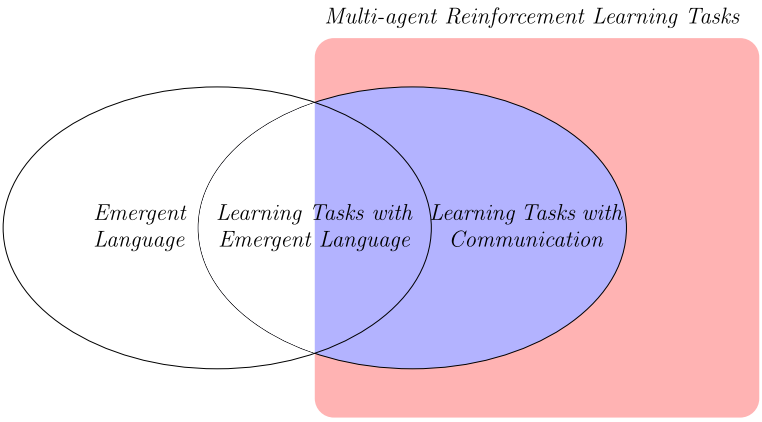


Fig. 1 An illustration depicting the scope of this survey. The focus of our survey is represented by the blue part

Furthermore, communication protocols, determining when and with whom to communicate, often leverage deep learning models to find the optimal choices that minimize communication overhead and yield more targeted communication. A multitude of works have been proposed to handle these subproblems inherent in Comm-MADRL. Most research works model only one or a few aspects of Comm-MADRL while selecting a default approach for other aspects. Given that the common goal of Comm-MADRL approaches is to design an effective and efficient communication protocol to improve agents' learning performance in the environment, the proposed Comm-MADRL approaches inevitably share similarities to some extent. Consequently, establishing a classification system for Comm-MADRL becomes crucial. Such a system would aid in categorizing critical elements like contributions, targeted problems, and learning objectives, from which we can compare and analyse existing Comm-MADRL approaches.

In the emergent language literature, numerous works employ various forms of the Lewis game, often referred to as referential games and operate under a cheap-talk setting [50], as highlighted in several surveys [10, 18].[5] In these games, a goal, often represented as a target location, an image, or a semantic concept, is given to a sender agent but remains unrevealed from a receiver agent. The receiver agent must then either identify the correct goal based on the sender's signaling [46, 51–57] or accomplish its single-agent task using the received signals (messages) [58, 59]. Research works in learning tasks with emergent language are grounded in a multi-agent environment where the joint actions of both sender and receiver agents impact environment transitions. Consequently, the learning tasks with emergent language literature considers multi-agent domain tasks [60–64], building on foundational concepts from MARL such as Dec-POMDPs or POSGs.

We further distinguish *explicit* versus *non-explicit* communication [6] in the literature of MADRL with communication. Explicit communication refers to communication through a set of messages separate from domain-level actions. Here, agents' action policies are

[5] In the emergent language research area, research works that do not adopt the cheap-talk setting but communicate through observable (domain-level) actions, are not included in our survey. Our survey focuses on explicit message transfer between agents.

influenced by both their observations and the messages they receive. Such messages, crucial for supporting agents' decision-making, are essential in both the training and execution phases. MADRL frameworks without explicit communication can still allow for communication through domain-level actions, such as the act of influencing the observations of one agent through the actions of another. Furthermore, without explicit communication, agents can transmit gradient signals, which facilitate centralized training (and decentralized execution) but are not utilized during execution phases. Specifically, in our survey, we focus on explicit and learnable communication.

Dec-POMDPs and POSGs are often extended to accommodate explicit communication. The communication can be integrated into the action set, adding a collection of communication acts alongside domain-level actions. Alternatively, a Dec-POMDP or a POSG can be extended to explicitly include a set of messages [6]. For instance, the POSG can be expanded with a (shared) message space $\mathcal{M}$, resulting in a POSG-Comm, defined as $\langle \mathcal{I}, \mathcal{S}, \rho^0, \{\mathcal{A}_i\}, P, \{\mathcal{O}_i\}, O, \{R_i\}, \mathcal{M} \rangle$, where all components remain unchanged except for the added message space $\mathcal{M}$. A Dec-POMDP-Comm can be defined as similar to the POSG-Comm with shared rewards. In both POSG-Comm and Dec-POMDP-Comm, action policies take into account both environmental observations and inter-agent messages. Research works in Comm-MADRL that expand upon a POSG or a Dec-POMDP can be seen in references such as [60, 62, 64–66].

2.3 Communication in recent surveys

Communication has attracted much attention in the field of multi-agent reinforcement learning (MARL). Previous surveys mentioning communication in MARL primarily focus on providing an overview of MARL's development. These surveys view communication as a subfield in MARL, and no extensive and substantial progress is reported. In an early survey, Stone and Veloso [13] classify MARL based on whether agents communicate and whether agents are homogeneous or not.[6] They view learnable communication as a future research opportunity. Busoniu et al. [15] consider communication as a means to negotiate action choices and select equilibrium in the research direction of explicit coordination, without further classifying communication. With the advancement of deep learning, MARL has gradually incorporated deep neural networks such that recent developments are dominated by multi-agent deep reinforcement learning (MADRL). In the MADRL context, Hernandez-Leal et al. [16], Nguyen et al. [67], and Papoudakis et al. [9] briefly review early Comm-MADRL methods, which have now become baselines in many recent works. Specifically, Hernandez-Leal et al. [16] use *learning communication* to denote a new branch in MADRL. Papoudakis et al. [9] consider communication as an approach to handle the non-stationary problem in MADRL, as agents can exchange information to stabilize their training. Compared to the aforementioned surveys, OroojlooyJadid and Hajinezhad [36] provide a more detailed review of Comm-MADRL, covering a significant number of existing works. They view communication as a way to solve cooperative MADRL problems but did not propose a categorization model for Comm-MADRL. Zhang et al. [68] and Yang et al. [20] review communication from a theoretical perspective. Their primary focus is on communication within networked multi-agent systems. In these systems, agents share

[6] Homogeneous agents have the same internal structure including goals, domain knowledge, and possible actions.

information through a time-varying network, aiming to reach consensus on learned value functions or policies. Despite this, no further classification of communication is made.

Two more recent surveys in MADRL, proposed by Gronauer and Diepold [17] and Wong et al. [69], focus on classifying existing works on communication. Gronauer and Diepold classify early research works in Comm-MADRL into Broadcasting, Targeted, and Networked communication, based on whether messages are received from all agents, a subset of agents, or a network of agents. Wong et al., similar to the survey of Papoudakis et al. [9], view communication as a method to address the issues of non-stationarity and partial observability. In the survey of Wong et al., research works on communication are categorized into three groups from a high-level perspective: communication as the primary learning goal, communication as an instrument to learn a specific task, and peer-to-peer teaching. However, they do not delve into how agents utilize communication to enhance learning. These surveys focus on limited aspects of communication, making their categorizations too narrow to distinguish recent works effectively, given the fact that many existing works share similar assumptions and conditions. To the best of our knowledge, only one survey [70] exclusively focuses on communication issues in MADRL. They review algorithms for communication and cooperation, including efforts to interpret languages developed through communication. Despite this, their survey mainly covers early models without proposing a categorization framework.

The literature has investigated communication from other perspectives. Shoham and Leyton-Brown [71] investigate communication from a game-theoretic perspective. They introduce several theories of communication in multi-agent systems, with the particular concern that agents can be self-motivated to convey information, driven by underlying incentives (e.g., the knowledge of game structure), or communicate in a pragmatic way analogous to human communication. Deep neural networks and deep reinforcement learning techniques have greatly widened the scope of language development in multi-agent systems. Lazaridou and Baroni [18] provide an extensive survey focused on *emergentlanguage*, aiming to establish effective human–machine communication. As highlighted in Sect. 2.2, the primary goal of emergent language research is to learn a human-like language from scratch. The goal of our survey is, however, to classify the literature on learning tasks with communication that aims at exploiting communication to accomplish multi-agent tasks.

In summary, existing surveys in Comm-MADRL lack coverage of the latest developments. These surveys also do not elaborate on the fact that communication itself is a combinatorial problem. Importantly, communication models engage with MADRL algorithms across various processes, including learning and decision-making. To effectively distinguish between existing Comm-MADRL approaches, it is crucial to analyze and classify them from a wider range of perspectives. In the following section, we delve into the field of Comm-MADRL through multiple dimensions, each linked to a unique research question pertinent to system design. These dimensions allow us to provide a fine-grained classification, highlighting the differences between Comm-MADRL approaches even within similar domains.

3 Learning tasks with communication in MADRL

In our survey, we consider explicit communication where action policies of agents are conditioned on communication that is learnable and dynamic, rather than static and predefined. Therefore, both the content of the messages and the chances of communication occurrences are subject to learning. As agents engage in multi-agent tasks, they learn domain-specific action policies and their communication protocols concurrently. As a result, *learning tasks with communication* becomes a joint learning challenge, where agents employ reinforcement learning to maximize environmental rewards and simultaneously utilize various machine learning techniques to develop efficient and effective communication protocols.

Learning tasks with communication in multi-agent deep reinforcement learning (Comm-MADRL) is a significant research problem, particularly as communication can lead to higher rewards. Numerous studies have emerged, developing effective and efficient Comm-MADRL systems, often sharing similarities. Our review begins with the seminal works such as DIAL [72], RIAL [72], and CommNet [47], and then expands to include the most relevant research works presented at major AI conferences and journals like AAMAS, AAAI, NeurIPS, and ICML, totaling 41 models in Comm-MADRL. To better distinguish among these models, we propose classifying them based on several dimensions in Comm-MADRL system design. These dimensions aim to comprehensively cover the current literature, allowing us to project the research works into a space where their similarities and differences become clear. We start by focusing on three key components of Comm-MADRL systems: problem settings, communication processes, and training processes. Problem settings encompass both communication-specific settings (e.g., communication constraints) and non-communication-specified settings (e.g., reward structures). Communication processes include common communication procedures, such as deciding whether to communicate and what messages to communicate. Training processes cover the learning of both agents and communication within MADRL. Based on the three key components, we identify and summarize 9 research questions that commonly arise in Comm-MADRL system design, corresponding to 9 dimensions as detailed in Table 1. These research questions and dimensions are designed to capture various aspects of Comm-MADRL, covering the learning objectives of agents and communication, the processes by which messages are generated, transmitted, integrated, and learned within the MADRL framework. We outline a systematic procedure for providing a guideline to effectively navigate through these dimensions when developing Comm-MADRL systems. The procedure allows us to organize the dimensions, demonstrate their relevance in system design, and guide the creation of customized Comm-MADRL systems in a step-by-step manner.

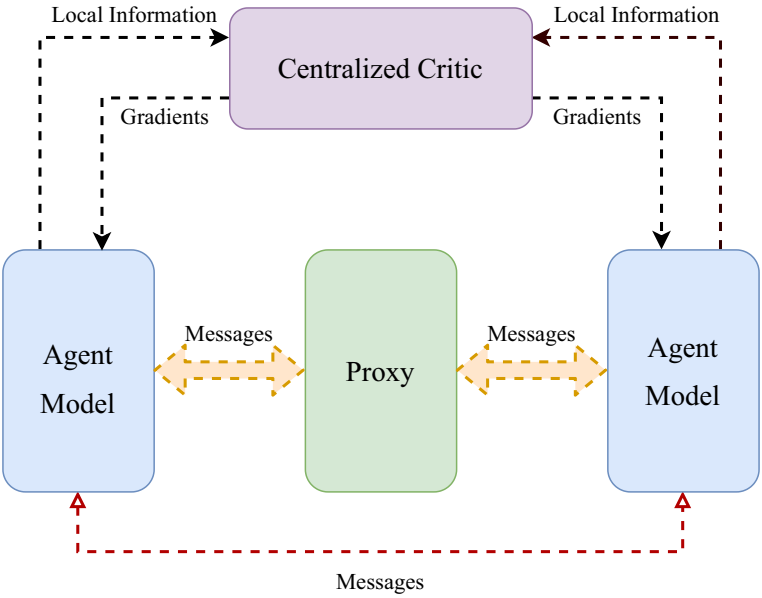
As outlined in Procedure 1, N reinforcement learning agents employ communication throughout their learning and decision-making. Initially, the learning objective for the N agents is set, defining rewards that induce cooperative, competitive, or mixed behaviors, as captured by dimension 1. We then consider potential communication-specified settings like limited resources, addressing the need for realistic scenarios as described in dimension 2. Dimension 3 identifies potential communicatees, determining the agents for messages to be received, which varies across domains. At each time step, agents decide when and with whom to communicate, as highlighted in dimension 4. The patterns of communication occurrences are structured like a graph, where links, either undirected or directed, aid information exchange. Subsequently, messages that encapsulate agents' understanding of the environment are generated and shared, relating to dimension 5. Given that agents often receive multiple messages, they must decide on how to combine these messages effectively.

Table 1 Proposed dimensions and associated research questions

Key components	Target questions	Dimensions	Index
Problem settings	What kind of behaviors are desired to emerge with communication?	Controlled goals	①
	How to fulfill realistic requirements?	Communication constraints	②
	Which type of agents to communicate with?	Communicatee type	③
Communication processes	When and how to build communication links among agents?	Communication policy	④
	Which piece of information to share?	Communicated messages	⑤
	How to combine received messages?	Message combination	⑥
	How to integrate combined messages into learning models?	Inner integration	⑦
Training processes	How to train and improve communication?	Learning methods	⑧
	How to utilize collected experience from agents?	Training Schemes	⑨

This process, crucial for integrating messages into their policies or value functions, is captured in dimensions 6 and 7. In cases of Comm-MADRL studies focusing on emergent language (i.e., learning tasks with emergent language), where messages are modeled as communicative acts emitted alongside domain-level actions, a specific rearrangement of the procedure is required. Here, messages are not observed by other agents until the next time step. Therefore, the processes outlined in dimensions 6 and 7 (lines 8 and 9) are moved to the front of those in dimension 4 (line 6). This rearrangement allows agents to combine and integrate messages from the previous time step before initiating new communication. As a result, agents make decisions and perform actions in the environment based not only on their environmental observations but also on information obtained from other agents (lines 10 and 11). During the training phase, experiences from both environmental interactions and inter-agent communication are utilized to train how agents will behave and communicate, i.e., agents' policies, value functions, and communication processes, as characterized in dimensions 8 and 9 (line 14).

In the following sections, we make an extensive survey on Comm-MADRL based on each dimension and classify the literature when we focus on a specific dimension. We finally provide a comprehensive table to frame recent works with the aid of the 9 dimensions.



Procedure 1 A guideline of Comm-MADRL systems

3.1 Controlled goal

With a given reward configuration, reinforcement learning agents are guided to achieve their designated goals and interests. As agents communicate in order to obtain higher rewards, the goal of communication and the goal of achieving domain-specific tasks are inherently aligned. The emergent behaviors of agents can be summarized into three types:

Table 2 The category of controlled goals

Types	Configurations	Methods
Cooperative	Global rewards	DIAL [72]; RIAL [72]; CommNet [47]; GCL [60]; MAGNet-SA-GS-MG [39]; MADDPG-M [40]; SchedNet [42]; Agent-entity graph [73]; VBC [30]; NDQ [74]; IMAC [65]; Gated-ACML [75]; Bias [62]; LSC [76]; Diff discrete [77]; I2C [78]; TMC [31]; GAXNet [79]; DCSS [63]; MAIC [32]
	Local rewards	BiCNet [49]; DGN [80]; IC3Net [48]; MD-MADDPG [41]; DCC-MD [81]; GA-Comm [82]; NeurComm [83]; IP [84]; ETCNet [85]; Variable-length coding [86]; AE-Comm [64]
	Global or local rewards	MS-MARL-GCM [87]; ATOC [38]; TarMAC [88]; IS [89]; HAMMER [90]; MAGIC [91]; FlowComm [92]; FCMNet [93]
Competitive	Conflict Rewards	IC3Net [48]; R-MACRL [66]
Mixed	Self-interested rewards	IC [61]; DGN [80]; TarMAC [88]; IC3Net [48]; NDQ [74]; LSC [76]; MAGIC [91]

cooperative, competitive, and mixed [22, 94], each corresponding to different reward configurations and goals. Notably, some Comm-MADRL methods have been tested in more than one benchmark environment to show their flexibility and scalability, where the reward configurations may vary [48, 76, 80, 88, 91]. Furthermore, a multi-agent environment may consist of both fixed opponents and teammates, which typically do not participate in communication. Therefore, we exclude fixed agents when identifying reward configurations. Consequently, we focus on (learnable) agents involved in communication and classify their behaviors that are desired to emerge, aligning them with associated reward configurations (summarized in Table 2).

Cooperative

In cooperative scenarios, agents have the incentive to communicate to achieve better team performance. Cooperative settings can be characterized by either a global reward that all agents share or a sum of local rewards that could be different among agents. Communication is usually used to promote cooperation as a team. Thus, in the literature, a team of agents can receive a global reward [30–32, 38–40, 42, 47, 60, 62, 63, 65, 72–79, 87–93], which does not account for the contribution of each agent. The agents can also receive local rewards, with designs to make the reward depend on teammates' collective performance [41, 48, 49, 80–82, 86, 87, 90], to penalize collisions [38, 80–82, 85, 89, 91, 92], or to share the reward with other agents for encouraging mutual cooperation [64, 83, 84].

There are a variety of cooperative environments where communication has shown performance improvements, from small-scale games to complex video games. In early works, Foerster et al. [72] developed two simple games, named Switch Riddle and MNIST Games, for their proposed models, DIAL and RIAL. Sukhbaatar et al. [47] used Traffic Junction for evaluating CommNet, which has become a popular testbed in recent works [48, 78, 82, 87–89, 91]. Among them, MAGIC [91] achieved higher performance on Traffic Junction with local rewards compared to two early works, CommNet [47], IC3Net [48], and one recent work, GA-Comm [82]. StarCraft [95–97] is another benchmark environment in cooperative MARL with relatively flexible settings. BiCNet [49] and MS-MARL-GCM [87] are evaluated on an early version of StarCraft [95]. Then, a new version of StarCraft, SMAC, has become popular in recent works [30–32, 65, 74, 93]. By controlling a team of agents, the cooperative goal in SMAC is to defeat enemies on easy, hard, and super hard maps. FCMNet [93] and MAIC are two recent works that surpass multiple communication methods and value decomposition methods (e.g., QMIX) on different maps. Google research football [98] is an even more challenging game with a physics-based 3D soccer simulator. Only MAGIC has reported performance on this platform with communication, and more investigations on this environment are needed. Compared to the above approaches in Comm-MADRL, ATOC [38] has been examined using a significantly larger number of learning agents in the predator–prey domain. Predator–prey is a grid world game with a long history in MARL. It has been developed with several versions [7, 99, 100], while still viewed as a standard test environment due to its flexibility and customizability. ATOC reports performance on this platform with continuous state and action spaces. In the subfield *learning tasks with emergent language*, cooperative scenarios are popularly used. They are mostly based on grid world or particle environments and have explicit role assignments, e.g., senders and receivers [60, 62–64].

Competitive

In case agents need to compete with each other to occupy limited resources, they are assigned competitive learning objectives. In some competitive games, such as zero-sum games, one player wins and the others lose and therefore rational agents do not have the incentive to communicate. Nevertheless, in other competitive scenarios where agents

Table 3 The category of communication constraints

Types	Subtypes	Methods
Unconstrained Communication		CommNet [47]; BiCNet [49]; MS-MARL-GCM [87]; ATOC [38]; DGN [80]; TarMAC [88]; MAGNet-SA-GS-MG [39]; MADDPG-M [40]; IC3Net [48]; MD-MADDPG [41]; DCC-MD [81]; Agent-Entity Graph [73]; GA-Comm [82]; LSC [76]; NeurComm [83]; IP [84]; I2C [78]; IS [89]; HAMMER [90]; MAGIC [91]; Flow-Comm [92]; GAXNet [79]; FCMNet [93]
Constrained Communication	Limited Bandwidth	RIAL [72]; DIAL [72]; GCL [60]; IC [61]; SchedNet [42]; VBC [30]; NDQ [74]; IMAC [65]; Gated-ACML [75]; Bias [62]; ETCNet [85]; Variable-length Coding [86]; TMC [31]; AE-Comm [64]; MAIC [32]
	Corrupted Messages	DIAL [72]; Diff Discrete[77]; DCSS [63]; $\mathfrak{R}$ -MACRL [66]

compete for long-term goals, communication can allow for low-level cooperation among agents before the (long-term) goals are achieved. Based on our observations, only one work, IC3Net [48], tests competitive settings and enables agents to compete for rewards.[7] IC3Net shows that competitive agents communicate only when it is profitable, e.g., before catching prey in the predator–prey domain. $\mathfrak{R}$-MACRL [66] considers communication from malicious agents to improve the worst-case performance. In $\mathfrak{R}$-MACRL, the whole environment is cooperative while agents learn to defend against malicious messages. Although the environment is cooperative, we classify this work under the competitive category as the learning goal between malicious agents and other agents is competitive.

Mixed

For a MAS where we care about self-interest agents, individual rewards can be designed and distributed to each agent [48, 74, 76, 80, 88, 91, 93]. Therefore, cooperative and competitive behaviors coexist during learning, which may show more complex communication patterns. Specifically, DGN [80] considers a game where each agent gets positive rewards by eating food but gets higher rewards by attacking other agents. However, being attacked will get a high punishment. With communication, agents can learn to share resources collaboratively rather than attacking each other. IC3Net [48], TarMAC [88] and MAGIC [91] are evaluated on a mixed version of Predator-prey, and agents learn to communicate only when necessary. NDQ [74] is examined in an independent search scenario, where two agents are rewarded according to their own goals, and shows that agents learn to not communicate in independent scenarios. IC [61] considers a scenario in which sender and receiver agents have different abilities to complete the goal. The sender agents have more vision but cannot clean obstacles, while receiver agents have limited vision but are able to clear obstacles. With communication, agents show collaborative behaviors to get higher rewards.

[7] IC3Net has been tested in several settings, including cooperative, competitive, and mixed scenarios, with different reward configurations.

3.2 Communication constraints

Practical concerns such as communication cost and noisy environment impair Comm-MADRL systems from embracing realistic applications more than simulations. This dimension, Communication Constraints, determines which type of communication concerns are handled in a Comm-MADRL system. We categorize recent works on this dimension into the following categories (summarized in Table 3).

Unconstrained communication

In this category, communication processes, including communication channels, the content and transmission of messages, and the decisions of whether to communicate or not, are not explicitly restricted. In principle, agents can communicate as much as information they can without any decision to disallow communication in order to prevent communication overhead [39, 41, 47, 49, 79, 80, 87–90, 93]. Specifically, several works consider blocking communication through predefined or learnable decisions of whether to communicate or not, while aiming to differentiate useful communicated information [38, 40, 48, 73, 76, 78, 81–84, 91, 92]. We also put those works under this category as they do not explicitly assume that communication is limited by cost.

Constrained communication

In this category, communication processes are explicitly constrained by cost or noise. Thus, agents need to utilize communication resources efficiently to promote learning. We further identify two practical concerns that have been considered in the literature.

- **Limited Bandwidth**. In this category, communication bandwidth is limited by channel capacity. Thus, communication needs to be used more efficiently, both in the number of times that agents can communicate and the size of communicated information. Early works focus on transmitting succinct messages to avoid communication overhead. RIAL and DIAL [72] are proposed to communicate very little information (i.e., a binary value or a real number) at every time step to reduce the bandwidth needed. MD-MADDPG [41] considers a fixed-size memory, which is shared by all agents. Agents communicate through the shared memory instead of ad hoc channels. VBC [30] and TMC [31] reduce communication costs by using predefined thresholds to filter unnecessary communication, and both show lower communication overhead. NDQ [74] cuts 80% of messages by ordering the distributions of messages according to their means and drops accordingly to prevent meaningless messages. MAIC [32] also cuts messages by examining several message pruning rates. In MAIC, messages are encoded to consider their respective importance. Sent messages are ordered and then pruned with a given pruning rate. IMAC [65] explicitly models bandwidth limitation as a constraint to optimization. An upper bound of the mutual information between messages and observations is derived according to bandwidth constraint, which turns out to minimize the entropy of messages. Then agents learn not only to maximize cumulative rewards but also to generate low-entropy messages. The number of agents to communicate can also be restricted to reduce the total amount of communication. SchedNet [42] considers a scenario of a shared channel together with limited bandwidth. Only a subset of agents are chosen to convey their messages according to their importance. Gated-ACML [75] learns a probabilistic gate unit to block messages transmitting between each agent and a centralized message coordinator, with the extra cost of learning optimal gates. Inspired by Gated-ACML and IMAC, ETCNet [85] puts constraints on the behaviors of deciding whether to send messages or not. A penalty term is added to the environ-

ment rewards, and an additional reinforcement learning algorithm is used to optimize the sending behaviors. Variable-length Coding [86] also utilizes a penalty term while encouraging short messages. When learning tasks with emergent language, symbolic languages are acquired for communication through a limited number of tokens. Therefore, we classify those works under limited bandwidth [60–62, 64].

- **Corrupted Messages**. In this category, messages transmitted among agents can be corrupted due to environmental noise or malicious intentions. DIAL [72] shows that during training, adding Gaussian noise to the communication channel can push the distribution of messages into two modes to convey different types of information. Diff Discrete [77] considers how to backpropagate gradients through a discrete communication channel (between 2 agents) with unknown noise. An encoder/channel/decoder system is modeled, where the encoder is used to discretize a real-valued signal into a discrete message to pass through the discrete communication channel, and the decoder is used to compute an approximation of the original signal. Later they show that the encoder/channel/decoder system is equivalent to an analog communication channel with additive noise. With the additional assumption that training is centralized, the gradient of the receiver with respect to real-value messages from the sender can be computed to allow backpropagation. DCSS [63] also considers a noisy setting. They prove that representing messages as one-hot vectors may not be optimal when the environment becomes noisy. Inspired by word embedding in the NLP field, they propose to generate a semantic representation of discrete tokens that are communicated among agents. The results show that such representation is robust in noisy environments and benefits human understanding of communication. Different from noisy environments, $\mathfrak{R}$-MACRL [66] assumes that an agent holds a malicious messaging policy, producing adversarial messages that can mislead other agents' action selections. Therefore, other agents need to prevent being exploited by learning a defense policy in order to filter the messages.

3.3 Communicatee type

Communicatee Type determines which type of agents are assumed to receive messages in a Comm-MADRL system. We found that in the literature, communicatee type can be classified into the following categories based on whether agents in the environment communicate with each other directly or not.

Agents in the MAS

In this category, the set of communicatees consists of agents in the environment, and they directly communicate with each other. Nevertheless, due to partial observability, agents may not be able to communicate with every agent in the MAS, and thus we further distinguish the types of communicatees as follows:

- **Nearby Agents**. In many Comm-MADRL systems, communication is only allowed between neighbors. Nearby agents can be defined as observable agents [79], agents within a certain distance [73, 76, 80] or neighboring agents on a graph [83]. GAXNet [79] labels observable agents and enables communication between them. DGN [80] limits communication within 3 closest neighbors while using a distance metric to find them. Agent-Entity Graph [73] also uses distance to measure whether agents are nearby or not. As long as two agents are close to each other, they will be allowed to communicate. LSC [76] enables agents within a cluster radius to decide whether to become

a leader agent. Then all non-leader agents in the same cluster will only communicate with the leader agent. NeurComm [83] and IP [84] preset a graph structure among agents built upon networked multi-agent systems. In both NeurComm and IP, communicatees are restricted to neighbors on the graph. MAGNet-SA-GS-MG [39] uses a pre-trained graph to limit communication and restricts communication on neighboring agents. Neighboring agents can also emerge during learning instead of being predetermined, as proposed in GA-Comm [82], MAGIC [91] and FlowComm [92], which explicitly learn a graph structure among agents. Specifically, in GA-Comm [82] and MAGIC [91], a central unit (e.g., GNN) learns a graph inside and coordinates messages based on the (complete) graph simultaneously. In this case, agents do not communicate with each other directly; instead, they communicate through a virtual agent who does not affect the environment. Therefore, we categorize these two works into the proxy category.

- **Other (Learning) Agents**. If nearby agents are not identified, the set of communicatees typically consists of other (learning) agents. Specifically, IC3Net [48] enables communication between learning agents and their opponents. Experiments indicate that these opponents eventually learn to not communicate to avoid being exploited. Some works assume explicit role assignments, i.e., senders and receivers. The role of the receiver can be taken by a disjoint set of agents separate from the senders [61–63] or by all other agents in the environment [60, 64]. In both cases, agents communicate with each other directly.

Proxy

A proxy is a virtual agent that plays an essential role (e.g., as a medium) in facilitating communication but does not directly affect the environment. Using a proxy as the communicatee means that agents will not directly communicate with each other, instead viewing the proxy as a medium, coordinating and transforming messages for specific purposes. MS-MARL-GCM [87] utilizes a master agent that collects local observations and hidden states from agents in the environment and sends a common message back to each of them. Similarly, HAMMER [90] employs a central proxy that gathers local observations from agents and sends a private message to each agent. MD-MADDPG [41] maintains a shared memory among agents, learning to selectively store and retrieve local observations from the memory. IMAC [65] defines a scheduler that aggregates encoded information from all agents and sends individual messages to each agent. These works primarily focus on how to encode messages through the proxy without determining whether to send or receive messages. By contrast, ATOC [38], Gated-ACML [101], GA-Comm [82] and MAGIC [91] are all designed for agents to decide whether to communicate with a message coordinator. In ATOC and Gated-ACML, each agent's decisions are made locally based on individual observations, with messages aggregated from nearby agents and from the entire MAS, respectively. Both GA-Comm and MAGIC develop a global communication graph, coupled with a graph neural network (GNN) to aggregate messages by weights and send new messages back to each agent, informing action selection in the environment.

Table 4 summarizes recent works on communication types in MAS. To illustrate these categories, we present an example of different communication methods used in a Comm-MADRL system in Fig. 2. The system consists of five agents and one proxy. Agent 3 is the nearby agent of Agent 1, while Agent 4 is the nearby agent of Agent 2. Agent 5 is out of the view range of Agents 1 and 2. If communication is limited to nearby agents, Agent 1 will communicate only with Agent 3, and Agent 2 will communicate only with Agent

Table 4 The category of communicatee type

Types	Subtypes	Methods
Agents in the MAS	Nearby Agents	DGN [80]; MAGNet-SA-GS-MG [39]; Agent-Entity Graph [73]; LSC [76]; NeurComm [83]; IP [84]; FlowComm [92]; GAXNet [79]
	Other Agents	DIAL [72]; RIAL [72]; CommNet [47]; GCL [60]; BiCNet [49]; IC [61]; TarMAC [88]; MADDPG-M [40]; IC3Net [48]; SchedNet [42]; DCC-MD [81]; VBC [30]; NDQ [74]; Bias [62]; Diff Discrete[77]; I2C [78]; IS [89]; ETCNet [85]; Variable-length Coding [86]; TMC [31]; AE-Comm [64]; DCSS [63]; R-MACRL [66]; MAIC [32]; FCMNet [93]
Proxy		MS-MARL-GCM [87]; ATOC [38]; MD-MADDPG [41]; IMAC [65]; GA-Comm [82]; Gated-ACML [75]; HAMMER [90]; MAGIC [91]

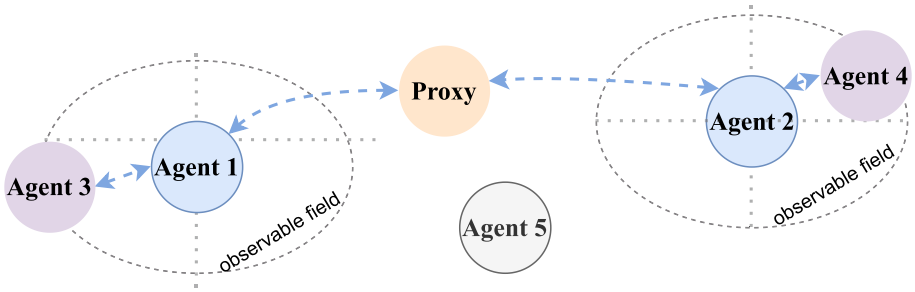


Fig. 2 Three communicatee types in the same system

4. However, if communication involves a proxy, all agents can send their messages to the proxy and receive coordinated messages.

3.4 Communication policy

Communication Policy determines when and with which agents (i.e., communicatees) to communicate in order to enable message transmission. A Communication Policy defines a set of communication actions, which can be modeled in different ways. For example, a communication action can be represented as a vector of binary values, where each value indicates whether communication with one of the other agents is allowed at a certain time step. These actions form communication links between pairs of agents, which can be represented as a communication graph among agents. In the literature, communication policies can be either predefined or learned, allowing communication with all other agents or only a subset of agents. Furthermore, communication policies can be centralized, controlling communication among all agents, or decentralized, enabling individual agents to control whether to communicate. Therefore, we first categorize the literature based on whether communication policies are predefined or learned. We find that in predefined communication policies, the literature often uses either full communication among agents, where the communication graph becomes complete, or a partial graph structure to incorporate constraints on communication policies. On the other hand, in learnable communication policies, we identify two distinct categories: individual control and global control. In individual control, communication policies are learned by each agent independently, whereas in global control, these policies are learned and implemented centrally, applying to all agents in Comm-MADRL systems. As a result, we have identified four subcategories within the dimension of communication policy: Full Communication, (Predefined) Partial Structure, Individual Control, and Global Control. These categorizations are summarized in Table 5.

We present examples of how agents form communication links in the four categories of communication policy, as illustrated in Fig. 3. Both Full Communication and Partial Structure rely on a predefined communication policy to determine communication actions. In contrast, Individual Control and Global Control involve the learning of a local communication policy and a global communication policy, respectively, to establish communication links between agents or a potential proxy. If a proxy is involved, it coordinates messages from agents choosing to communicate through this proxy. The categories and their associated research works are introduced as follows:

Table 5 The category of communication policy

Types	Subtypes	Methods
Predefined	Full Communication	DIAL [72]; RIAL [72]; CommNet [47]; GCL [60]; BiCNet [49]; MS-MARL-GCM [87]; TarMAC [88]; MD-MADDPG [41]; DCC-MD [81]; IMAC [65]; Diff Discrete[77]; IS [89]; Variable-length Coding [86]; HAMMER [90]; AE-Comm [64]; R-MACRL [66]; FCMNet [93]
	Partial Structure	IC [61]; DGN [80]; MAGNet-SA-GS-MG [39]; Agent-Entity Graph [73]; VBC [30]; NDQ [74]; Bias [62]; NeurComm [83]; IP [84]; TMC [31]; GAXNet [79]; DCSS [63]; MAIC [32]
Learnable	Individual Control	ATOC [38]; MADDPG-M [40]; IC3Net [48]; Gated-ACML [75]; LSC [76]; I2C [78]; ETCNet [85]
	Global Control	SchedNet [42]; GA-Comm [82]; MAGIC [91]; FlowComm [92]

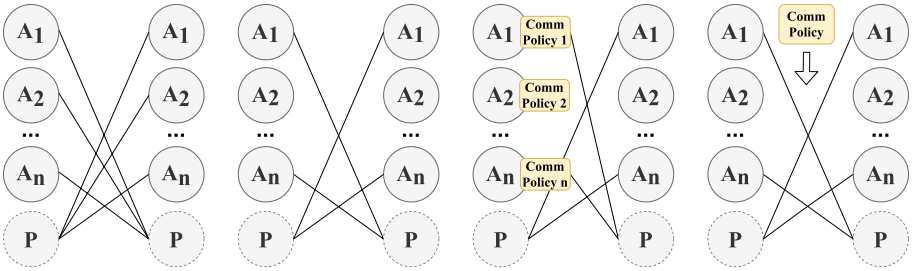


Fig. 3 Four types of communication policy with agents (shown as A) in the environment and a possible proxy (shown as P)

Full communication

In this category, every pair of agents is connected so that messages are transmitted in a broadcast manner. Full communication can be regarded as a fully connected graph, often used in early works on Comm-MADRL. DIAL [72], RIAL [72], CommNet [47], and BiCNet [49] learn a communication protocol which connect all agents together. Inspired by BiCNet, FCMNet [93] uses multiple RNNs to link all agents together with different sequences, allowing agents to benefit from communication flow from various directions. In contrast, Diff Discrete [77] and Variable-length Coding [86] focus on two-agent cases but do not learn to block messages from each other. TarMAC [88] and IS [89] learn meaningful messages while using a broadcast way to share messages, thus still adhering to full communication. DCC-MD [81] and $\mathfrak{R}$-MACRL [66] introduce a strategy to drop out received messages without specifying whether to send messages. Specifically, DCC-MD drops out messages with a fixed probability to reduce input dimensions, and $\mathfrak{R}$-MACRL learns to drop out adversary messages through a defense policy. In Comm-MADRL methods like IMAC [65], MS-MARL-GCM [87] and HAMMER [90], a central proxy that receives local observations or encoded messages is always connected with agents in the MAS. In addition, GCL [60] and AE-Comm [64] learn a language grounded on discrete tokens among agents, where all agents have the capability to send and receive messages.

(Predefined) Partial structure

In this category, the communication between agents is captured by a predetermined partial graph to reduce overall communication. Then, each agent communicates with a limited number of agents within the MAS, rather than with every agent. NeurComm [83] and IP [84] operate in a networked multi-agent environment, randomly generating a communication network while maintaining a fixed average number of connections per agent during the learning process. DGN [80], MAGNet-SA-GS-MG [39], and GAXNet [79] restrict communication to a certain proximity of agents. The Agent-Entity Graph [73] employs a pre-trained graph to capture agent relationships. Comm-MADRL approaches like VBC [30], NDQ [74], TMC [31], and MAIC [32] utilize handcrafted thresholds or pruning rates to limit communication opportunities. In IC [61], Bias [62], and DCSS [63], disjoint sets of agents are designated as either senders or receivers, facilitating unidirectional communication from senders to receivers only.

Individual control

In this category, each agent actively and individually determines whether to communicate with other agents, implicitly forming a graph structure. A common method employed in Comm-MADRL studies within this category is a learnable gate mechanism, which aids agents in making the decision to communicate. For instance, IC3Net [48] and ATOC [38]

Table 6 The category of communicated messages

Types	Methods
Existing Knowledge	DIAL [72]; RIAL [72]; CommNet [47]; GCL [60]; BiCNet [49]; MS-MARL-GCM [87]; IC [61]; DGN [80]; TarMAC [88]; MAGNet-SA-GS-MG [39]; MADDPG-M [40]; IC3Net [48]; MD-MADDPG [41]; SchedNet [42]; DCC-MD [81]; Agent-Entity Graph [73]; VBC [30]; NDQ [74]; IMAC [65]; GA-Comm [82]; Gated-ACML [75]; Bias [62]; LSC [76]; Diff Discrete[77]; I2C [78]; ETCNet [85]; Variable-length Coding [86]; TMC [31]; HAMMER [90]; MAGIC [91]; FlowComm [92]; AE-Comm [64]; GAXNet [79]; DCSS [63]; R-MACRL [66]; MAIC [32]; FCMNet [93]
Imagined Future Knowledge	ATOC [38]; NeurComm [83]; IP [84]; IS [89]

use a gate mechanism that enables agents to decide whether to broadcast their messages, in a deterministic and probabilistic manner, respectively. ETCNet [85] also implements a gate unit but limits the overall probability of message-sending behaviors. If a proxy, such as a message coordinator, is present, Gated-ACML [75] introduces a learning mechanism for each agent to decide whether to communicate with the proxy, as opposed to direct communication with other agents. Diverging from the gate function approach, I2C [78] allows each agent to unilaterally decide on communication with other agents, based on evaluating the impact of those agents on its own policy. LSC [76] allows each group of agents, defined by a specific radius, to compare their weights in order to elect a leader. This system then facilitates communication from each group to their respective leaders and from leader to leader. Notably, the leader agent in this model is not considered a proxy, as it still directly interacts with the environment.

Global control

In this category, a globally shared communication policy is learned, providing more complete control over the communication links between agents. SchedNet [42] employs a global scheduler that limits the number of agents allowed to broadcast their messages, thereby reducing overall communication. FlowComm [92] learns a directed graph among agents, enabling unilateral or bilateral communication between them. Similarly, GA-Comm [82] and MAGIC [91] develop an undirected and a directed graph for communication, respectively. These Comm-MADRL systems incorporate an additional message coordinator to coordinate and transform messages sent by the agents.

3.5 Communicated messages

After establishing communication links among agents through a communication policy, agents should determine which specific information to communicate. This information can derive from historical experiences, intended actions, or future plans, enriching the messages with valuable insights. Consequently, the communicated information can expand the agents' understanding of the environment and enhance the coordination of their behaviors. In the dimension of communicated messages, an important consideration is whether the communication includes future information, such as intentions and plans. This kind of information, being inherently private, often requires an (estimated) model of the

environment to effectively simulate and generate conjectured intentions and plans. Accordingly, we categorize recent studies in this dimension into two categories, as summarized in Table 6.

Existing knowledge

In this category, agents share their knowledge of the environment (e.g., past observations), previous movements, or policies to assist other agents in selecting actions. As historical information accumulates, agents use a low-dimensional encoding of their knowledge as messages to reduce communication overhead. Notably, the RNN family (e.g., LSTM and GRU) is commonly used as an encoding function, capable of selectively retaining and forgetting historical observations [32, 41, 47–49, 78, 82, 87, 88, 91–93], action-observation histories [49, 72], or action-observation-message histories [61, 62]. When a proxy is present, messages are generated and transformed from agents to the proxy, and then from the proxy to agents. Thus, local observations can either be encoded [41, 65, 75, 82, 91] or directly sent [87, 90] to the proxy. The proxy, after gathering these local (encoded) observations, can generate a unified message for all agents [87], or individualized messages for each agent [41, 65, 75, 82, 90, 91]. Both methods provide a message containing global information, relieving agents from the task of combining multiple received messages. In Comm-MADRL systems without a proxy, messages are sent directly to each agent. Specifically, in MADDPG-M [40], agents communicate local observations without an encoding of them. On the other hand, DIAL and RIAL [72] encode past observations, actions, and current observations as messages. BiCNet [49] encodes both local observations of each agent and a global view of the environment. Other research works employ various methods such as simple feed-forward networks [42, 77, 85, 86], MLP [30, 31, 39, 66], autoencoders [81], CNNs [80], RNNs [32, 47, 48, 78, 88, 92, 93], or GNNs [73, 76] to encode local observations as messages. Furthermore, agents can communicate more specific information, such as in GAXNet [79], where agents coordinate their local attention weights, integrating hidden states from neighboring agents. Messages can also be modeled as random variables, as seen in NDQ [74], where messages are drawn from a multivariate Gaussian distribution to maximize expressiveness by maximizing mutual information between messages and receivers' action selection. In learning tasks with emergent language, agents often communicate goal-related information, such as the goal's location [60–64].

Imagined future knowledge

In this context, Imagined Future Knowledge refers to aspects such as intended actions [38], policy fingerprints (i.e., action probabilities in a given state) [83, 84], or future plans [89]. Since intentions are related to the current environment state, recent works often combine intended actions with local observations to produce more relevant messages. The concept of future plans extends this idea further by utilizing an approximated model of the environment and the behavior models of other agents. This approach enables the generation of a sequence of possible future observations and actions [89]. Such knowledge is shared among agents, allowing the receivers to consider the potential future outcomes of the senders' actions.

3.6 Message combination

When agents receive more than one message, current works often aggregate all received messages to reduce the input for the action policy. Message Combination determines how to integrate multiple messages before they are processed by an agent's internal model. If a

Table 7 The category of message combination

Types	Methods
Equally Valued	DIAL [72]; RIAL [72]; CommNet [47]; GCL [60]; IC [61]; MADDPG-M [40]; IC3Net [48]; SchedNet [42]; VBC [30]; NDQ [74]; Bias [62]; Diff Discrete[77]; IS [89]; ETCNet [85]; Variable-length Coding [86]; FlowComm [92]; AE-Comm [64]; DCSS [63]
Unequally Valued	BiCNet [49]; MS-MARL-GCM [87]; ATOC [38]; DGN [80]; TarMAC [88]; MAGNet-SA-GS-MG [39]; MD-MADDPG [41]; DCC-MD [81]; Agent-Entity Graph [73]; IMAC [65]; GA-Comm [82]; Gated-ACML [75]; LSC [76]; NeurComm [83]; IP [84]; I2C [78]; TMC [31]; HAMMER [90]; MAGIC [91]; GAXNet [79]; R-MACRL [66]; MAIC [32]; FCMNet [93]

proxy is involved, each agent receives already coordinated and combined messages from the proxy, eliminating the need for further message combination. If no proxy is presented, each agent independently determines how to combine multiple messages. Since communicated messages encode the senders' understanding of the learning process or the environment, some messages can be more valuable than others. As shown in Table 7, recent works in the dimension of message combination are categorized based on how agents prioritize received messages.

Equally valued

In this category, messages received by agents are treated without preference, meaning they are assigned equal weights or simply no weights at all. Without having preferences, agents can concatenate all messages, ensuring no loss of information, though it may significantly expand the input space for the action policy [40, 42, 60, 61, 72, 74, 77, 85, 86, 89]. Recent research involving concatenated messages typically represent the sent messages either as single values [61, 72, 85, 86] or as short vectors [40, 42, 60, 74, 77, 89]. Alternatively, messages can be combined by averaging [30, 47, 48] or summing [92], under the assumption that messages from different agents have the same dimension. In some cases, particularly in two-agent scenarios, no explicit preferences are assigned to messages [62–64].

Unequally valued

In this category, messages are assigned distinct preferences, which potentially impose differences on sender agents. DCC-MD [81] and TMC [31] use handcrafted rules to prune received messages. In DCC-MD, each received message can be dropped out with a certain probability. TMC stores the received messages and checks whether they are expired or not within a preset time window. Only valid messages are integrated into an agent's model. Instead of using fixed rules, $\mathfrak{R}$-MACRL [66] learns a gate unit to decide whether to use a received message. An attention mechanism can

Table 8 The category of inner integration

Types	Methods
Policy-level	CommNet [47]; GCL [60]; MS-MARL-GCM [87]; ATOC [38]; MAGNet-SA-GS-MG [39]; IC3Net [48]; MD-MADDPG [41]; SchedNet [42]; IMAC [65]; GA-Comm [82]; Gated-ACML [75]; Diff Discrete[77]; IP [84]; I2C [78]; IS [89]; ETCNet [85]; Variable-length Coding [86]; HAMMER [90]; Flow-Comm [92]; GAXNet [79]; R-MACRL [66]
Value-level	DIAL [72]; RIAL [72]; DGN [80]; DCC-MD [81]; VBC [30]; NDQ [74]; LSC [76]; TMC [31]; MAIC [32]
Policy- and Value-level	BiCNet [49]; IC [61]; TarMAC [88]; MADDPG-M [40]; Agent-Entity Graph [73]; Bias [62]; NeurComm [83]; MAGIC [91]; AE-Comm [64]; DCSS [63]; FCMNet [93]

also be learned to assign weights to received messages and then combine them, rather than filtering messages out, as seen in research works [32, 39, 73, 88]. Moreover, a neural network can aggregate received messages into a single message or a low-dimensional vector, which implicitly imposes preferences on messages during the mapping. Feedforward neural networks [65, 75, 90], CNNs [80], LSTMs (or RNNs) [38, 41, 49, 78, 79, 83, 87, 93], and GNNs [76, 82, 84, 91] have been used as aggregators. Among them, GNNs utilize a learned graph structure of agents and assign different weights to neighboring agents.

3.7 Inner integration

Inner Integration determines how to integrate (combined) messages into an agent's learning model, such as a policy or a value function. In most existing literature, messages are viewed as additional observations. Agents take messages as extra input to a policy function, a value function, or both. Thus, in the dimension of inner integration, we classify recent works into categories based on the learning model that is used to integrate messages. These categories are summarized in Table 8.

Policy-level

By exploiting information from other agents, each agent will no longer act independently. Policies can be learned through policy gradient methods like REINFORCE, as seen in studies [47, 48, 60, 82, 87], which collect rewards during episodes and train the policy models at the end of episodes. Moreover, the Comm-MADRL approaches that utilize actor-critic methods [38, 39, 41, 42, 65, 66, 75, 77–79, 84–86, 89, 90, 92] assume that a critic model (i.e., a Q-function) guides the learning of an actor model (i.e., a policy network).

Value-level

In this category, a value function incorporates messages as input, and a policy is derived by selecting the action with the highest Q-value. Most works in this category employ DQN-like methods to train their value functions [30–32, 72, 74, 76, 80, 81]. Specifically, Comm-MADRL approaches like VBC [30], NDQ [74], TMC [31], and MAIC [32] are based on value decomposition methods in cooperative scenarios (with global rewards). These methods involve learning to decompose a joint Q-function.

Policy- and value-level

Integrating messages using both a policy function and a value function typically relies on actor-critic methods. In Comm-MADRL approaches within this category, received

Table 9 The assumptions behind different learning methods

Types	Assumptions
Fully differentiable	The messages or the communication actions are generated by a differentiable function and thus backpropagation is used everywhere
Supervised learning	True labels (or the ground truth) are assumed to be given or defined to guide the learning of communication policy or messages
Reinforcement learning	Environment rewards or self-defined rewards are used to update communication policy or messages incrementally
Regularizers	Regularizations such as entropy inspired from information theory are added to agents' optimization objectives to regularize the learning of communication

Table 10 The category of learning methods

Types	Methods
Differentiable	GCL [60]; DIAL [72]; CommNet [47]; BiCNet [49]; MS-MARL-GCM [87]; DGN [80]; TarMAC [88]; MAGNet-SA-GS-MG [39]; MD-MADDPG [41]; DCC-MD [81]; Agent-Entity Graph [73]; VBC [30]; GA-Comm [82]; Diff Discrete[77]; NeurComm [83]; IP [84]; IS [89]; Variable-length Coding [86]; TMC [31]; MAGIC [91]; FlowComm [92]; GAXNet [79]; DCSS [63]; FCMNet [93]
Supervised	DCSS [63]; ATOC [38]; Gated-ACML [75]; I2C [78]; R-MACRL [66]
Reinforced	GCL [60]; RIAL [72]; IC [61]; MADDPG-M [40]; IC3Net [48]; SchedNet [42]; LSC [76]; ETCNet [85]; HAMMER [90]
Regularized	NDQ [74]; IMAC [65]; Bias [62]; AE-Comm [64]; MAIC [32]

messages can be treated as extra inputs for both the actor and critic models [49, 63, 73]. Alternatively, messages can be combined with local observations to generate new internal states, which are then shared with both the actor and critic models [40, 61, 62, 64, 83, 88, 91, 93].

3.8 Learning methods

Learning methods determine which type of machine learning techniques is used to learn a communication protocol. The learning of communication is at the center of modern Comm-MADRL and can benefit from the advancements in the machine learning field. If proper assumptions about communication are made, such as being able to calculate the derivatives with respect to the message generator function and the communication policy, then the training of communication can be integrated into the overall learning process of agents. This integration allows for the use of fully differentiable methods for backpropagation. Other machine learning techniques, including reinforcement learning, supervised learning, and regularizations, can also be utilized to incorporate our requirements and available ground truth into the learning of communication, each carrying respective assumptions. The assumptions used in the literature are summarized in Table 9. For instance, supervised methods require defining true labels for communication (e.g., the correct information to share or the right agents to communicate with). In contrast, reinforced methods

use rewards as learning signals. Regularized methods, which use neither true labels nor rewards, employ an additional learning objective by using regularizers, such as minimizing the entropy of messages to reduce stochasticity. Therefore, we classify recent works based on how they differ in the learning of communication (summarized in Table 10).

Differentiable

In this category, communication is learned and improved by backpropagating gradients from agent to agent. When the communication policy is predefined, such as in full communication [41, 47, 49, 60, 72, 77, 81, 86–89, 93] or by communicating with a subset of agents [30, 31, 39, 63, 73, 79, 80, 83, 84], agents learn the content of messages through backpropagation. Several recent studies [60, 63, 82, 91, 92] address the issue of non-differentiable communication actions by utilizing gradient estimators like Gumbel-softmax [102], which replaces non-differentiable samples with a differentiable approximation during training, albeit requiring additional parameter tuning. Specifically, both GCL [60] and DCSS [63] employ a differential message function in their approaches. Additionally, GCL integrates auxiliary rewards, and DCSS utilizes labeled messages for training communication policies. Thus, they are categorized under the Differentiable category, each additionally aligning with the Reinforced and Supervised categories respectively. Freed et al. [77] propose an alternative method, Diff Discrete, to address the challenge of continuous messages versus discrete channels. This method models message transmitting as an encoder/channel/decoder system, where the receiver decodes the messages and reconstructs the original signals. These reconstructed signals enable the calculation of derivatives with respect to the sender, allowing gradients to be sent back to the sender.

Supervised

In this category, additional efforts need to be made to define the true label for when and what information to communicate. ATOC [38] and Gated-ACML [75] use the difference in Q-values between actions chosen with and without a message to define a label of communication actions. If the difference exceeds a threshold, the message is deemed valuable, indicating a high probability of sending it; otherwise, the probability is 0. This process sets up a classification task to decide whether to communicate. Similarly, I2C [78] trains a classifier to determine communication but relies on the causal effect between two agents, using a threshold to tag effective communication. $\mathfrak{R}$-MACRL [66] learns a classifier to identify malicious messages, using the status of a message (malicious or not) as a label. DCSS [63] learns message content by using a small dataset that maps observations to desired communication symbols. In DCSS, the gradient from the supervised loss is added to the policy loss, leading agents to use communication that aligns with the grounding data and enables high task performance.

Reinforced

In this category, reinforcement learning is utilized to train communication in addition to the learning of action policies. RIAL [72] and HAMMER [90] focus on learning the content of messages through reinforcement learning, without addressing the decision of whether to communicate. In GCL [60], auxiliary rewards are used for predicting goals and consolidating symbols, facilitating the development of a compositional language for communication. IC [61] employs the difference in outcomes from using and not using communication on action policies as rewards. Maximizing the rewards can enhance the influence of communication on the receivers' action policies. Other studies [40, 42, 48, 76, 85] consider both the learning of communication content and the decision to communicate. Notably, MADDPG-M [40] suggests using intrinsic rewards to train the communication policy instead of relying solely on environmental rewards. ETCNet [85] shapes environmental rewards by introducing a penalty term to discourage unnecessary communication.

Regularized

Regularized methods are used to reduce redundant information in communication [32, 65, 74]. NDQ [74] calculates a lower bound of the mutual information between received messages and the receivers' action selection. This approach suggests that messages can be optimized to decrease the uncertainty in the action-value functions of the receivers. IMAC [65] establishes an upper bound on the mutual information between messages and the senders' observations, and minimizing this upper bound helps agents send messages with lower uncertainty. MAIC [32] employs an estimated model of teammates and aims to maximize the mutual information between teammates' actions and hidden variables from this model. This model then guides the encoding of messages, resulting in tailored communications for different agents. Bias [62] focuses on the long-term impact of messages on agents' decision-making to enhance signaling and listening effectiveness. AE-Comm [64] adopts an autoencoder to learn a low-dimensional encoding of observations.

3.9 Training schemes

This dimension focuses on how to utilize the collected experiences (such as observations, actions, rewards, and messages) of agents to train their action policies and communication architectures in a Comm-MADRL system. Agents can train their models in a fully decentralized manner using only their local experience. Alternatively, when global information is accessible, the experiences of all agents can be collected to centrally train a single (centralized) model that controls all agents. However, each approach has inherent challenges. Fully decentralized learning must cope with a non-stationary environment due to the changing and adapting behaviors of agents, while fully centralized learning faces the complexities of joint observation and policy spaces. As a balanced solution, Centralized Training and Decentralized Execution (CTDE) [72, 103] has emerged as a popular training schemes in MADRL. CTDE approaches allow agents to learn their local policies using guidance from central information. Therefore, in the dimension of training schemes, we categorize recent works based on how agents' experiences are collected and utilized, as detailed in Table 11.

Centralized learning

As shown in Fig. 4a, experiences are gathered into a central unit, then learning to control all agents. Based on our observations, recent works on Comm-MADRL usually do not assume a central controller.

Fully decentralized learning

As illustrated in Fig. 4b, in fully decentralized learning, experiences are collected individually by each agent, and they undergo independent training processes. Recent works in this category often employ actor-critic based methods for each agent [39, 40, 66, 73, 81, 83, 84]. Specifically, decentralized learning has gained much attention in *learning tasks with emergent language*, as it most closely resembles language learning in nature [61, 62, 64].

Centralized training and decentralized execution

In CTDE approaches, the experiences of all agents are collectively used for optimization. Gradients derived from the joint experiences of agents guide the learning of local policies. However, once training is complete, only the policies are needed and gradients can be discarded, facilitating decentralized execution. When agents are assumed to be homogeneous, meaning they have identical sensory inputs, actuators, and model structures, they can share parameters. Parameters sharing reduces the overall number of parameters, potentially

Table 11 The category of training schemes

Types	Subtypes	Methods
Fully Decentralized Learning		IC [61]; MAGNet-SA-GS-MG [39]; MADDPG-M [40]; DCC-MD [81]; Agent-Entity Graph [73]; Bias [62]; NeurComm [83]; IP [84]; AE-Comm [64]; R-MACRL [66]
Centralized Training and Decentralized Execution	Individual Parameters	MS-MARL-GCM [87]; SchedNet [42]; IMAC [65]; Gated-ACML [75]; GAXNet [79]; DCSS [63]
	Parameter Sharing	DIAL [72]; RIAL [72]; CommNet [47]; GCL [60]; BiCNet [49]; ATOC [38]; DGN [80]; TarMAC [88]; IC3Net [48]; VBC [30]; NDQ [74]; GA-Comm [82]; LSC [76]; Diff Discrete[77]; I2C [78]; ETCNet [85]; Variable-length Coding [86]; TMC [31]; HAMMER [90]; MAGIC [91]; FlowComm [92]; MAIC [32]; FCMNet [93]
	Concurrent	MD-MADDPG [41]; IS [89]

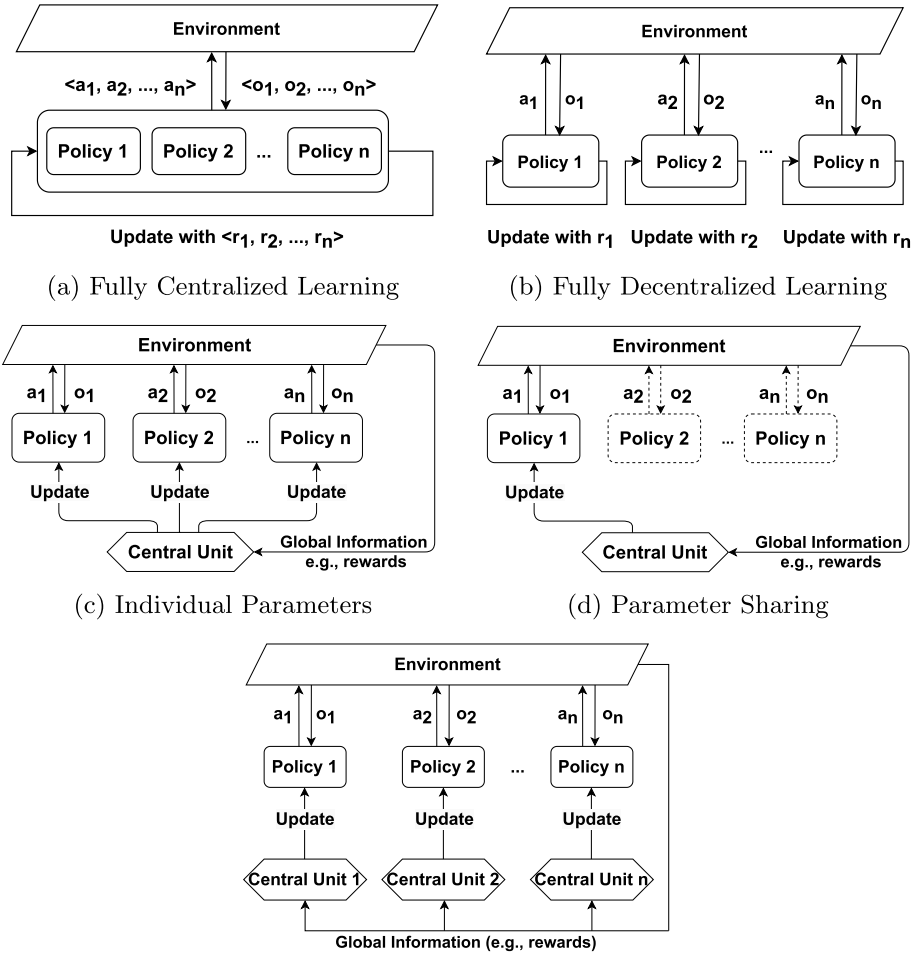


Fig. 4 Five types of training schemes

enhancing learning efficiency compared to training in separate processes. Despite sharing parameters, agents can still exhibit distinct behaviors because they are likely to receive different observations at the same time step. Based on these considerations, recent works in this field can be further divided into the following subcategories.

- **Independent Policies**. In this category, each local policy is trained with its own set of learning parameters. A central unit collects experiences from all agents to provide global information and guidance, such as gradients, as depicted in Fig. 4c. The training of the entire system can employ policy gradient algorithms (e.g., using REINFORCE) [87], or actor-critic methods [42, 63, 65, 75, 79].
- **Parameter Sharing**. In this category, all local policies (or local value functions) utilize a shared set of parameters, as illustrated in Fig. 4d. Commonly used algorithms in this scenario include DQN-like algorithms, actor-critic methods, and pol-

icy gradient algorithms with REINFORCE. When employing a DQN-like algorithm, a shared local Q-function, which processes each agent's individual experience, is learned collectively across agents [72, 76, 80]. Additionally, DQN-based methods can be integrated with value decomposition models (e.g., QMIX [27]) in cooperative environments, which enable learning from factorized rewards (value functions) [30–32, 74]. In the case of actor-critic methods, a shared actor (i.e., policy model) is trained using all individual experiences, supported by gradient guidance from a central critic [38, 49, 77, 78, 85, 86, 88, 90–93]. Policy gradient with REINFORCE can alternatively be used, requiring the collection of sampled rewards over episodes [47, 48, 60, 82].

- **Concurrent**. In scenarios where storing all experiences in a central unit is not feasible, agents can alternatively create backups of all experiences, with the assumption that they are able to observe other agents' actions and observations. The concurrent approaches differ inherently from fully decentralized learning. In CTDE with concurrent approaches, each agent maintains an individual set of policy parameters and receives the guidance from a local unit that collects global information (with additional assumptions on observability), as depicted in Fig. 4e. Concurrent CTDE often employs actor-critic methods, where each agent has its own central critic to guide its local actor (policy) [41, 89].

3.10 Possible relations of dimensions

We have introduced 9 dimensions for Comm-MADRL and identified a range of categories within each dimension. It is crucial to consider the potential interdependencies among these dimensions. We realize that the dimensions do not inherently depend on one another based on the criteria used for classifying the literature. However, specific implementations of Comm-MADRL systems may create dependencies between dimensions. For instance, limited bandwidth constraints (defined in the communication constraints dimension) can be realized by setting a limited number of times to communicate, rendering the full communication category (within the communication policy dimension) infeasible. This scenario illustrates how the dimensions of communication constraints (Sect. 3.2) and communication policy (Sect. 3.4) become interdependent due to specific implementations. Another example about communicated messages shows that the classification criteria we used do not depend on each other. During implementation, a proxy (in the communicatee type dimension) or corrupted message constraints (in the communication constraints dimension) may change the value of message content. However, we categorize communicated messages as Existing Knowledge or Imagined Future Knowledge, based on whether future knowledge is simulated and utilized. This classification criterion is not inherently linked to a specific type of communicatee or communication constraint. Thus, the dimensions of communicatee type (Sect. 3.3) and communication constraints (Sect. 3.2) are independent from the viewpoint of classification criteria. Consequently, the proposed categories and dimensions effectively encapsulate the literature from their unique perspectives.

Table 12 The notations of all categories

Dimensions	Notations
Controlled Goals (CG)	$\mathcal{C}_{oo}$: Cooperative; $\mathcal{C}_{om}$: Competitive; $\mathcal{M}$: Mixed
Communication Constraints (CC)	$\mathcal{U}$: Unconstrained Communication;$\mathcal{L}_b$: Limited Bandwidth; $\mathcal{C}_m$: Corrupted Messages
Communicatee Type (CT)	$\mathcal{N}_a$: Nearby Agents; $\mathcal{A}$: Other (Learning) Agents; $\mathcal{P}$: Proxy
Communication Policy (CP)	$\mathcal{F}_c$: Full Communication; $\mathcal{P}_s$: Predefined (Partial) Structure; $\mathcal{I}_c$: Individual Control; $\mathcal{G}_c$: Global Control
Communicated Messages (CM)	$\mathcal{E}$: Existing Knowledge; $\mathcal{I}$: Imagined Future Knowledge
Message Combination (MC)	$\mathcal{V}_e$: Equally Valued; $\mathcal{V}_u$: Unequally Valued
Inner Integration (II)	$\mathcal{P}_l$: Policy-level; $\mathcal{V}_l$: Value-level; $\mathcal{PV}$: Policy-level & Value-level
Learning Methods (LM)	$\mathcal{D}$: Differentiable; $\mathcal{S}_p$: Supervised; $\mathcal{R}_e$: Reinforced; $\mathcal{R}_g$: Regularized
Training Schemes (TS)	$\mathcal{CL}$: Centralized Learning; $\mathcal{DL}$: Decentralized Learning; $\mathcal{CTDE}_{ip}$: CTDE with Individual (Policy) Parameters; $\mathcal{CTDE}_{ps}$: CTDE with Parameter Sharing; $\mathcal{CTDE}_c$: Concurrent CTDE

4 Findings, discussions, and research directions

In this section, we discuss the trend of the current literature and provide our observations and findings based on the proposed dimensions and categorizations. We also dive into the dimensions and suggest possible future research directions.

4.1 Findings and discussions

To provide a more comprehensive overview of the literature, we have utilized the proposed 9 dimensions to categorize existing works, thereby creating an extensive table. For ease of reference, we introduce notations for these dimensions and their associated categories in Table 12. These notations are subsequently employed to categorize research works in Table 13. In Table 13, research works are sorted based on their publication or archival dates (e.g., on arXiv). Our proposed 9 dimensions offer different perspectives for analyzing and comparing recent works in the field of Comm-MADRL. Through these dimensions and categories, we have observed several intriguing findings.

- In the dimension of Controlled Goals, recent research has focused on various cooperative settings, together with a few mixed scenarios. Communication in non-cooperative multi-agent tasks, however, has not been extensively explored. In such (non-cooperative) environments, the goals of different agents may conflict. In the emergent language literature, Noukhovitch et al. [46] have investigated how communication emerges between sender and receiver agents when they exhibit different levels of competitiveness, ranging from full cooperation to full competition. The results reveal that both sender and receiver agents can obtain higher rewards through communication when the level of competition is not high. However, their research primarily focuses on a simplified game without considering state transitions. The effectiveness of communication in MARL tasks with large state spaces, particularly in partial competitive settings where agents can still gain mutual benefits through low-level cooperation, remains an area for

Table 13 An overview of recent works in Comm-MADRL

Methods	CG	CC	CT	CP	CM	MC	II	LM	TS
DIAL [72]	$\mathcal{C}_{oo}$	$\mathcal{L}_b$+$\mathcal{C}_m$	$\mathcal{A}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{V}_l$	$\mathcal{D}$	$\mathcal{CTDE}_{ps}$
RIAL [72]	$\mathcal{C}_{oo}$	$\mathcal{L}_b$	$\mathcal{A}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{V}_l$	$\mathcal{R}_e$	$\mathcal{CTDE}_{ps}$
CommNet [47]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{A}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{P}_l$	$\mathcal{D}$	$\mathcal{CTDE}_{ps}$
GCL [60]	$\mathcal{C}_{oo}$	$\mathcal{L}_b$	$\mathcal{A}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{P}_l$	$\mathcal{D}$+$\mathcal{R}_e$	$\mathcal{CTDE}_{ps}$
BiCNet [49]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{A}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{PV}$	$\mathcal{D}$	$\mathcal{CTDE}_{ps}$
MS-MARL-GCM [87]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{P}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{P}_l$	$\mathcal{D}$	$\mathcal{CTDE}_{ip}$
ATOC [38]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{P}$	$\mathcal{I}_c$	$\mathcal{I}$	$\mathcal{V}_u$	$\mathcal{P}_l$	$\mathcal{S}_p$	$\mathcal{CTDE}_{ps}$
IC [61]	$\mathcal{C}_{oo}$	$\mathcal{L}_b$	$\mathcal{A}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{PV}$	$\mathcal{R}_e$	$\mathcal{DL}$
DGN [80]	$\mathcal{C}_{od}/\mathcal{M}$	$\mathcal{U}$	$\mathcal{N}_a$	$\mathcal{P}_s$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{V}_l$	$\mathcal{D}$	$\mathcal{CTDE}_{ps}$
TarMAC [88]	$\mathcal{C}_{od}/\mathcal{M}$	$\mathcal{U}$	$\mathcal{A}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{PV}$	$\mathcal{D}$	$\mathcal{CTDE}_{ps}$
MAGNet-SA-GS-MG [39]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{N}_a$	$\mathcal{P}_s$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{P}_l$	$\mathcal{D}$	$\mathcal{DL}$
MADDPG-M [40]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{A}$	$\mathcal{I}_c$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{PV}$	$\mathcal{R}_e$	$\mathcal{DL}$
IC3Net [48]	$\mathcal{C}_{od}/\mathcal{C}_{om}/\mathcal{M}$	$\mathcal{U}$	$\mathcal{A}$	$\mathcal{I}_c$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{P}_l$	$\mathcal{R}_e$	$\mathcal{CTDE}_{ps}$
MD-MADDPG [41]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{P}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{P}_l$	$\mathcal{D}$	$\mathcal{CTDE}_c$
SchedNet [42]	$\mathcal{C}_{oo}$	$\mathcal{L}_b$	$\mathcal{A}$	$\mathcal{G}_c$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{P}_l$	$\mathcal{R}_e$	$\mathcal{CTDE}_{ip}$
DCC-MD [81]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{A}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{V}_l$	$\mathcal{D}$	$\mathcal{DL}$
Agent-Entity Graph [73]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{N}_a$	$\mathcal{P}_s$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{PV}$	$\mathcal{D}$	$\mathcal{DL}$
VBC [30]	$\mathcal{C}_{oo}$	$\mathcal{L}_b$	$\mathcal{A}$	$\mathcal{P}_s$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{V}_l$	$\mathcal{D}$	$\mathcal{CTDE}_{ps}$
NDQ [74]	$\mathcal{C}_{od}/\mathcal{M}$	$\mathcal{L}_b$	$\mathcal{A}$	$\mathcal{P}_s$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{V}_l$	$\mathcal{R}_g$	$\mathcal{CTDE}_{ps}$
IMAC [65]	$\mathcal{C}_{oo}$	$\mathcal{L}_b$	$\mathcal{P}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{P}_l$	$\mathcal{R}_g$	$\mathcal{CTDE}_{ip}$
GA-Comm [82]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{P}$	$\mathcal{G}_c$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{P}_l$	$\mathcal{D}$	$\mathcal{CTDE}_{ps}$
Gated-ACML [75]	$\mathcal{C}_{oo}$	$\mathcal{L}_b$	$\mathcal{P}$	$\mathcal{I}_c$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{P}_l$	$\mathcal{S}_p$	$\mathcal{CTDE}_{ip}$
Bias [62]	$\mathcal{C}_{oo}$	$\mathcal{L}_b$	$\mathcal{A}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{PV}$	$\mathcal{R}_g$	$\mathcal{DL}$
LSC [76]	$\mathcal{C}_{od}/\mathcal{M}$	$\mathcal{U}$	$\mathcal{N}_a$	$\mathcal{I}_c$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{V}_l$	$\mathcal{R}_e$	$\mathcal{CTDE}_{ps}$
Diff Discrete[77]	$\mathcal{C}_{oo}$	$\mathcal{C}_m$	$\mathcal{A}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{P}_l$	$\mathcal{D}$	$\mathcal{CTDE}_{ps}$
NeurComm [83]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{N}_a$	$\mathcal{P}_s$	$\mathcal{I}$	$\mathcal{V}_u$	$\mathcal{PV}$	$\mathcal{D}$	$\mathcal{DL}$
IP [84]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{N}_a$	$\mathcal{P}_s$	$\mathcal{I}$	$\mathcal{V}_u$	$\mathcal{P}_l$	$\mathcal{D}$	$\mathcal{DL}$
I2C [78]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{A}$	$\mathcal{I}_c$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{P}_l$	$\mathcal{S}_p$	$\mathcal{CTDE}_{ps}$
IS [89]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{A}$	$\mathcal{F}_c$	$\mathcal{I}$	$\mathcal{V}_e$	$\mathcal{P}_l$	$\mathcal{D}$	$\mathcal{CTDE}_c$
ETCNet [85]	$\mathcal{C}_{oo}$	$\mathcal{L}_b$	$\mathcal{A}$	$\mathcal{I}_c$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{P}_l$	$\mathcal{R}_e$	$\mathcal{CTDE}_{ps}$
Variable-length Coding [86]	$\mathcal{C}_{oo}$	$\mathcal{L}_b$	$\mathcal{A}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{P}_l$	$\mathcal{D}$	$\mathcal{CTDE}_{ps}$
TMC [31]	$\mathcal{C}_{oo}$	$\mathcal{L}_b$	$\mathcal{A}$	$\mathcal{P}_s$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{V}_l$	$\mathcal{D}$	$\mathcal{CTDE}_{ps}$
HAMMER [90]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{P}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{P}_l$	$\mathcal{R}_e$	$\mathcal{CTDE}_{ps}$
MAGIC [91]	$\mathcal{C}_{od}/\mathcal{M}$	$\mathcal{U}$	$\mathcal{P}$	$\mathcal{G}_c$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{PV}$	$\mathcal{D}$	$\mathcal{CTDE}_{ps}$
FlowComm [92]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{N}_a$	$\mathcal{G}_c$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{P}_l$	$\mathcal{D}$	$\mathcal{CTDE}_{ps}$
AE-Comm [64]	$\mathcal{C}_{oo}$	$\mathcal{L}_b$	$\mathcal{A}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{PV}$	$\mathcal{R}_g$	$\mathcal{DL}$
GAXNet [79]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{N}_a$	$\mathcal{P}_s$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{P}_l$	$\mathcal{D}$	$\mathcal{CTDE}_{ip}$
DCSS [63]	$\mathcal{C}_{oo}$	$\mathcal{C}_m$	$\mathcal{P}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_e$	$\mathcal{PV}$	$\mathcal{D}$+$\mathcal{S}_p$	$\mathcal{CTDE}_{ip}$
R-MACRL [66]	$\mathcal{C}_{om}$	$\mathcal{C}_m$	$\mathcal{A}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{P}_l$	$\mathcal{S}_p$	$\mathcal{DL}$
MAIC [32]	$\mathcal{C}_{oo}$	$\mathcal{L}_b$	$\mathcal{A}$	$\mathcal{P}_s$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{V}_l$	$\mathcal{R}_g$	$\mathcal{CTDE}_{ps}$
FCMNet [93]	$\mathcal{C}_{oo}$	$\mathcal{U}$	$\mathcal{A}$	$\mathcal{F}_c$	$\mathcal{E}$	$\mathcal{V}_u$	$\mathcal{PV}$	$\mathcal{D}$	$\mathcal{CTDE}_{ps}$

$a + b$ denotes that the research work considers categories a and b simultaneously in the environment. a/b denotes that the research work has been examined in multiple categories but in separate environments or settings

further exploration. Moreover, in non-cooperative settings, agents may be motivated to deceive or manipulate the communication channel to mislead others. The notion of *trust* in multi-agent systems introduces the possibility of establishing a truthful communication protocol [104, 105]. Agents could assess the reliability of opponents and defend against malicious messages. Additionally, agents might evaluate interactions of opponents with other agents to determine their reputations, which could influence the priorities of communicating with other agents.

- In the dimension of Communication Constraints, many existing works do not account for communication constraints, which may limit their applicability in realistic scenarios that have such limitations. For instance, transmitting messages in a large multi-agent system across long distances can result in delays, losses, or even be infeasible. Communication might be asynchronous, requiring several time steps for information exchange. These factors introduce new challenges to Comm-MADRL systems, such as validating previously sent messages and integrating messages from different time steps. Moreover, if communication resources are limited due to budget or capacity constraints, agents must decide how to allocate these resources effectively, especially when their goals vary. Conveying too much information might benefit others while sacrificing the agent's own learning opportunities. The concept of *fairness*, which has been extensively studied in multi-agent systems, focuses on developing fair solutions for resource allocation. The ideas of maximizing the utility of worse-off agents and decreasing the difference in utilities between agents in the fairness study can be utilized to distribute communication resources equally. For instance, agents with lower utilities could be allotted more resources to facilitate their communication with others.
- In the dimension of Communicatee Type, the concept of a proxy is utilized to facilitate message coordination. When global observability is available, a proxy often considers all agents within the environment. This proxy can be particularly effective and targeted by utilizing the independence among agents, coordinating messages among only a subset of agents as necessary.
- In the dimension of Communication Policy, current works often assume a binary communication action regarding whether or not to communicate with other agents (or a specific agent). However, communication actions can be more fine-grained and descriptive. For instance, agents might opt to send only a portion of their messages due to uncertainty or lack of confidence. Additionally, a communication action could be defined more specifically, such as *communicate with others if the budget exceeds a predetermined threshold*. Thus, a communication policy can encompass a variety of communication actions, tailored to align with human heuristics and specific system requirements.
- In the dimension of Communicated Messages, various methods have been proposed to utilize the existing knowledge of agents for message generation. Some existing works consider incorporating agents' intentions or future plans. However, intentions or future plans may lead to catastrophic errors due to insufficient understanding of the underlying (transition) dynamics. Model-based Reinforcement Learning (RL) could assist agents in making more accurate predictions about future situations, thereby enabling the agents to communicate information with more certainty regarding upcoming changes. Additionally, current literature often assumes that messages are conveyed as single values or vectors. In contrast, modern devices allow for more complex formats, such as graphs and logical expressions. These formats can convey a substantial amount of knowledge or facts concisely, facilitating fast coordination. However, the challenge lies

in effectively encoding and decoding complex information structures, which requires more sophisticated learning signals.

- In the dimension of Message Combination, as messages often contain information related to each agent's individual experiences, goals, etc., many recent works consider the varying importance of these messages. These research works mostly rely on attention mechanisms to impose weights on received messages. Furthermore, agents can incorporate their prior knowledge or preferences about other agents' capabilities into these weights, enhancing the relevance and effectiveness of message combination.
- In the dimension of Inner Integration, many recent works have focused on integrating messages into the policy model. This trend is likely due to the growing interest in policy-based methods, particularly actor-critic algorithms, within the field of MADRL, where significant advancements have been achieved. Given that neural networks typically feature a hierarchical structure, there is potential for agents to effectively integrate messages into different layers. This approach would allow for considering varying levels of abstraction, potentially enhancing the decision-making process.
- In the dimension of Learning Methods, the learning process for communication typically requires instantaneous feedback from agents who receive and act upon messages. This feedback could be in the form of gradient information or changes in the policies or rewards of the receiving agents. However, obtaining instantaneous feedback from other agents might not always be feasible in real-time decision-making systems. Despite this challenge, agents can still observe changes in the environment and their rewards to self-evaluate the effectiveness of their communication. This self-evaluation process enables agents to update and learn their communication protocols over time.
- In the dimension of Training Schemes, parameter sharing combined with centralized training and decentralized execution is widely adopted in Comm-MADRL to reduce the number of learning parameters. However, accessing other agents' memories and parameters might raise privacy concerns. On the other hand, fully decentralized learning presents significant challenges and remains a key research area in MARL. In fully decentralized learning, agents have limited knowledge about the environment and must deal with non-stationarity, a problem that intensifies with an increasing number of agents. Nonetheless, Comm-MADRL can benefit from advancements in MARL, potentially leading to the development of novel training paradigms that better balance knowledge sharing, privacy, and learning efficiency.

Based on the proposed dimensions, we have identified a range of findings and potential issues in the field of Comm-MADRL. Among these issues, achieving fully decentralized learning and self-evaluated communication protocols remains a significant challenge. This difficulty arises because each agent has access only to their own data collected from the environment, adding complexity to message evaluation without the help of other agents. Decentralized action policies and self-evaluated communication protocols, however, could be advantageous in areas like Electronic Commerce [106], Networks [107], and Blockchain [108], where synchronizing knowledge and information among users or agents can be computationally demanding. Another open question involves how to effectively communicate using more complex message formats and implement efficient training methods, potentially leading to more sophisticated communication architectures. Importantly, advancements in multi-agent systems and multi-agent reinforcement learning can significantly contribute to the progress of Comm-MADRL.

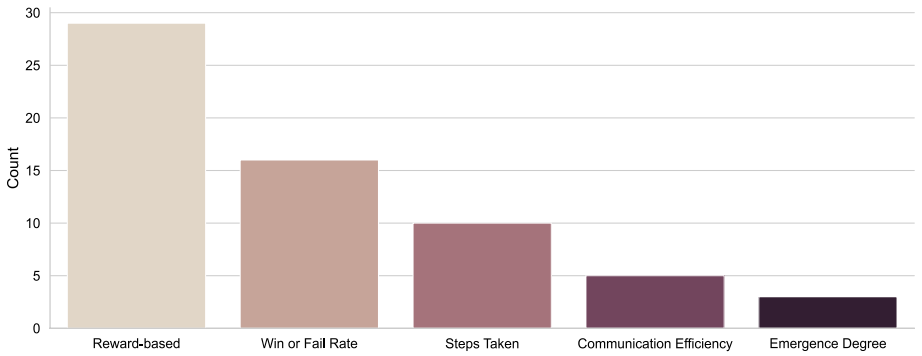


Fig. 5 The Statistics of Evaluation Metrics in existing Comm-MADRL systems

In addition to these findings, the evaluation metrics used in Comm-MADRL research are of significant interest. It is noteworthy that existing works have been evaluated across various platforms and games, employing different metrics to assess performance. Crucially, Comm-MADRL studies often use varying settings of experiments, such as the number of agents or the use of parameter sharing. These settings can make it challenging to fairly compare the relative strengths and limitations of algorithms based on their performance outcomes [109]. We have identified four evaluation metrics commonly used in Comm-MADRL studies as follows:

- *Reward-based*: This metric employs the converged return or average rewards per episode or time step to demonstrate the profit gained by agents.
- *Win or Fail Rate*: This metric calculates the percentage that agents achieve their goal or fail the game during learning. It is often used in episodic tasks.
- *Steps Taken*: This metric evaluates the number of time steps learned to reach the goal. It is often used in episodic tasks and essential in scenarios where time efficiency is key.
- *Communication Efficiency*: This metric evaluates how much communication resource has been used, such as the frequency of communication between agents.
- *Emergence Degree*: Originating from the field of emergent language, this metric evaluates and detects the emergence of language [44, 110]. It is often used in learning tasks with emergent language. *Positive signaling* and *positive listening* are two common approaches. Positive signaling measures the correlation between a message and the sender's observation or intended action. Positive listening assesses the impact of an observed message on the receiver's beliefs or behavior.

We have analyzed the number of times that the above performance metrics are used in existing Comm-MADRL studies, as illustrated in Fig. 5. It is shown that the metric of communication efficiency has not been extensively used in the literature, requiring further investigation into the use of communication resources in Comm-MADRL approaches. The Emergence Degree metric, intended to measure whether a language is emergent, is primarily utilized in emergent language studies. Nonetheless, this metric can also yield significant insights for other Comm-MADRL systems. By analyzing the correlation between communication and the observations and behaviors of both senders and receivers, we could obtain a deeper understanding and explanation of communication for Comm-MADRL.

In the next section, inspired by the proposed dimensions, we demonstrate the potential for discovering new ideas through our survey. We identify several possible research directions that jointly explore multiple dimensions, aiming to bridge the gaps in current works.

4.2 Research directions

Comm-MADRL is a young but rapidly enlarging field. There are still lots of possibilities to develop new Comm-MADRL systems. Our proposed dimensions encapsulate several aspects of Comm-MADRL, from which we can identify new research directions. Therefore, we showcase four research directions motivated by leveraging the possible combinations of dimensions and the extensions of corresponding categories. We also point out further challenges for Comm-MADRL.

4.2.1 Multimodal communication

A versatile robot can hear by sound sensors, read text or talk with human partners. Intelligent agents may be surrounded by different data sources and act based on multimodal input. By jointly considering the dimensions of communicatee type and communicated messages, we can imagine a fertile scenario where communication is not limited to images or handcrafted features but encompasses multimodal data, such as speech, videos, and text from humans or domestic robots, to prosper applications like smart home. To the best of our knowledge, existing works in Comm-MADRL do not consider communicating multimodal data or encoding them. Recent works often use encoded images as messages, which only cover visually-based applications. Therefore, we believe exploring multimodal communication represents a promising research direction and introduces several challenges that need to be addressed. In multimodal communication, agents have to coordinate heterogeneous modalities and encode various types of information into messages. A possible solution is to use separate channels to communicate specific modalities, while agents must decide on the right channel to communicate and merge data from different channels. A more efficient way is to learn a joint representation of multimodal observations and communicate on one channel. Due to the progress of Multimodal machine learning [111], we can bring ideas from this area to equip agents with the ability to create a single representation of multimodal data. Nevertheless, it is unclear how the solutions from Multimodal machine learning can be extended to multi-agent reinforcement learning. Poklukar et al. [112] propose learning an aligned representation from multiple modalities, although their tests are conducted in a single-agent reinforcement learning task. The multi-agent scenarios, however, may need to consider the individual abilities and preferences of different agents. For example, a voice-activated agent may favor voice data for interaction, while a monitoring agent may only access video data. Therefore, in multi-agent settings, agents need to align their individual preferences regarding multimodality when learning a joint representation of the multimodal data. Another crucial technical issue is how to represent multimodal messages in low-dimensional vectors without losing essential information from each modality, as Comm-MADRL systems often consider reducing communication costs. Eventually, we expect the progress of multimodal communication will benefit human-agent interaction and diverse communicating agents.

The emergence of new research works would introduce new categories under each dimension. For example, with developments in multimodal communication, we can extend the categories of communicated messages with speech, image, text, and video data.

Nevertheless, our proposed dimensions can be adaptive and robust to cover new Comm-MADRL research in this direction.

4.2.2 Structural communication

Through the Internet, electronic devices like routers can process and transmit information. On social media, chatbots can be community members [113, 114], to engage in conversations with users and share information/opinions. In those large-scale multi-agent systems [115, 116], agents may belong to different groups, where their relationships can be complicated. For example, local area networks create boundaries of communication and interaction between devices. Chatbots may not be able to reach some users because of limited permission or the lack of friendship relations. These restricted connectivities among agents require more efficient usage of communication structure. Therefore, we think that the research direction focusing on structural communication opens up possibilities for enabling communication among a larger number of agents. In the current literature in Comm-MADRL, ATOC [38] and LSC [76] have investigated communication with multiple groups, where agents can only communicate with other agents who belong to the same group. In both approaches, different groups may share common member agents, i.e., bridge agents, which are used to enable information to flow from group to group. However, communication through bridge agents is not targeted and each agent unconsciously shares their information with other groups. In terms of the dimension of controlled goals, agents may have individualized goals and require collaboration with a specific set of agents. Therefore, an important future direction of structural communication is to send critical information and opinions to target agents. For example, agent 1 may observe the goal location of agent 2 while they belong to different groups. If agent 3 happens to be a common friend of agents 1 and 2, agent 1 can actively send the goal information to agent 2 with the help of agent 3. If communication is costly and information is private, agents need to make thoughtful decisions about which bridge agents to be used to find the shortest and safe path to reach targeted agents. At the same time, bridge agents need to agree on the communication path to transmit information successfully. If a complex and hierarchical friendship network is identified, another important question is how to prioritize and schedule different communication paths to make communication fluent. Regarding communicated messages, agents need to build a common protocol with targeted agents so that information can be encoded and decoded successfully. As a result, agents can more actively utilize the communication structure among agents to achieve better collaboration and agreements.

4.2.3 Robust centralized unit

Robustness has been widely considered in the field of reinforcement learning [117, 118], where an agent needs to cope with disturbances in learning in order to achieve a robust policy that can generalize under changes in training/test data. In MARL, agents' policies can be sensitive to environmental noise or malicious intentions of opponents, and thus robust policies are required [119, 120]. With communication, opponents may produce malicious messages, implying adversary intentions. Preventing malicious messages is important in non-cooperative settings as adversary agents may manipulate communication to

achieve their own goals at the expense of other agents' benefits. Existing works on Comm-MADRL, such as $\mathfrak{R}$-MACRL [66], have investigated how to detect adversary information and reconstruct original messages. However, as we discussed in the dimensions of communicatee type and training paradigm, proxy and critics are often centralized and gather information from all agents. Robustness becomes essential for these centralized units as all agents involved in communication can be misled by polluted feedback, for example, incorrect gradient signals from critics or malicious messages from a proxy. Moreover, malicious messages can easily spread through the (centralized) proxy. Therefore, we think building a robust centralized unit is a promising and underdeveloped direction for safe communication in MADRL, where proxies and critics need to avoid communication being exploited by adversaries or affected by harmful environmental changes. By considering the dimension of communication policy, sender agents can learn a versatile communication policy. For example, the communication policy can be defined to select different encoding protocols for different groups of agents, in case malicious agents may easily find a solution to cheat on a specific encoding protocol. Besides, as malicious or noisy messages can be hidden in the centralized proxy, it is important to figure out which messages are malicious and how to reconstruct the original messages. Nonetheless, developing robust centralized units is vital for reliable and protected Comm-MADRL systems.

4.2.4 Learning tasks with emergent language

In this survey, we have identified the intersection between learning tasks with communication and emergent language in the field of MADRL, which we have called learning tasks with emergent language. We also observed that there is only a limited number of research works concerning this sub-area learning tasks with emergent language, which learns a language while achieving a MADRL task. We believe this area can be further expanded and investigated, by considering several dimensions proposed by our survey. First, the communicated messages, as we discussed earlier, can be encoded into more complex symbolic formats, such as graphs or logical expressions. Existing works in the field only learn how to communicate through atomic symbols or a combination [60, 62, 63]. However, it is important to learn the relation between symbols. For example, symbol A is on the left of symbol B. Those messages can express facts about what agents know or conjecture. Therefore, receivers can quickly adapt their behaviors by successfully decoding the messages. The important question is how to learn both encoding and decoding with complex expressions of messages, which can have a significant number of possibilities. The senders should also properly encapsulate their knowledge and the receivers should reason on the messages correctly. In addition, how complex symbolic formats can emerge in non-cooperative settings is an interesting but unexplored research area. What's more, the combination of complex messages will not be as easy as handling single values or vectors. Therefore, learning together with complicated communication is still challenging.

4.2.5 Further challenges

In the field of Comm-MADRL, there are further challenges. For instance, the design of neural network architectures plays a critical role in performance and communication. A deeper neural network may be effective in some domains while failing in other domains. For example, LSTM is effective in capturing history information while may require much time to train the parameters [121, 122], which could greatly slow down the learning in

tasks with high complexity. The choice of architectures and fine-tuning hyperparameters are significant problems of Comm-MADRL. With communication, another crucial issue is the explainability of communicated messages. Emergent language has made a step towards human-like language. However, whether machines communicate in a human-like way and can learn a human-interpretable language is still unclear. A great number of existing works regarding learning tasks with communication seek hidden, deep, and obscure codes for messages [47, 48, 78, 91], which still need to be further interpreted and understood.

5 Conclusions

Our survey proposes to classify the literature based on 9 dimensions. These dimensions constitute the basis of designing Comm-MADRL systems. We further categorize existing works under each dimension, where readers can easily compare research works from a unique perspective. Based on those dimensions, we also observe findings through the trend of the literature and identify new research directions by filling the gap among recent works. Our survey concludes that while the number of works in Comm-MADRL is notable and represents significant achievements, communication can be more fruitful and versatile to incorporate non-cooperative settings, heterogeneous players, and many more agents. Agents can communicate information not only from raw image inputs or handcrafted features but also from diverse data sources such as voice and text. Furthermore, we can explore novel metrics to better understand the contribution of communication to the overall learning process. Ultimately, Comm-MADRL can benefit from the MARL community and take advantage of good solutions from MARL.

References

1. Shalev-Shwartz, S., Shammah, S., & Shashua, A. (2016). Safe, multi-agent, reinforcement learning for autonomous driving. *CoRR*arXiv:1610.03295.
2. Vinyals, M., Rodríguez-Aguilar, J. A., & Cerquides, J. (2011). A survey on sensor networks from a multiagent perspective. *The Computer Journal, 54*(3), 455–470.
3. Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research, 32*(11), 1238–1274.
4. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T. P., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., & Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature, 550*(7676), 354–359.
5. Brown, N., & Sandholm, T. (2019). Superhuman AI for multiplayer poker. *Science, 365*(6456), 885–890.
6. Oliehoek, F. A., & Amato, C. (2016). *A concise introduction to decentralized POMDPs*. Berlin: Springer.

7. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems 30. Annual conference on neural information processing systems 2017, December 4–9, 2017, Long Beach, CA, USA*, (pp. 6379–6390).
8. Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., & Whiteson, S. (2018). Counterfactual multi-agent policy gradients. In S. A. McIlraith & K. Q. Weinberger (Eds.), *Proceedings of the thirty-second AAAI conference on artificial intelligence, (AAAI-18), the 30th innovative applications of artificial intelligence (IAAI-18), and the 8th AAAI symposium on educational advances in artificial intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2–7, 2018*, (pp. 2974–2982). New York.
9. Papoudakis, G., Christianos, F., Rahman, A., & Albrecht, S. V. (2019). Dealing with non-stationarity in multi-agent deep reinforcement learning. *CoRR*arXiv:1906.04737.
10. Zaïem, M. S., & Bennequin, E. (2019). Learning to communicate in multi-agent reinforcement learning: A review. *CoRR*arXiv:1911.05438.
11. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444.
12. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529–533.
13. Stone, P., & Veloso, M. M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots, 8*(3), 345–383.
14. Panait, L., & Luke, S. (2005). Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems, 11*(3), 387–434.
15. Busoniu, L., Babuska, R., & De Schutter, B. (2008). A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38*(2), 156–172.
16. Hernandez-Leal, P., Kartal, B., & Taylor, M. E. (2019). A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems, 33*(6), 750–797.
17. Gronauer, S., & Diepold, K. (2021). Multi-agent deep reinforcement learning: A survey. *Artificial Intelligence Review, 55*, 1–49.
18. Lazaridou, A., & Baroni, M. (2020). Emergent multi-agent communication in the deep learning era. *CoRR*arXiv:2006.02419.
19. Hansen, E. A., Bernstein, D. S., & Zilberstein, S. (2004). Dynamic programming for partially observable stochastic games. In McGuinness, & D. L., Ferguson, G. (Eds.) *Proceedings of the nineteenth national conference on artificial intelligence, sixteenth conference on innovative applications of artificial intelligence, July 25–29, 2004, San Jose, California, USA* (pp. 709–715).
20. Yang, Y., & Wang, J. (2020). An overview of multi-agent reinforcement learning from game theoretical perspective. *CoRR*arXiv:2011.00583.
21. Tan, M. (1993). Multi-agent reinforcement learning: Independent versus cooperative agents. In P. E. Utgoff (Ed.) *Machine learning, proceedings of the tenth international conference, University of Massachusetts, Amherst, MA, USA, June 27–29, 1993*, (pp. 330–337).
22. Matignon, L., Laurent, G. J., & Fort-Piat, N. L. (2012). Independent reinforcement learners in cooperative Markov games: A survey regarding coordination problems. *The Knowledge Engineering Review, 27*(1), 1–31.
23. Claus, C., & Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. In J. Mostow, & C. Rich (Eds.) *Proceedings of the fifteenth national conference on artificial intelligence and tenth innovative applications of artificial intelligence conference, AAAI 98, IAAI 98, July 26–30, 1998, Madison, Wisconsin, USA*, (pp. 746–752).
24. Tampuu, A., Matiisen, T., Kodelja, D., Kuzovkin, I., Korjus, K., Aru, J., Aru, J., & Vicente, R. (2015). Multiagent cooperation and competition with deep reinforcement learning. *CORR*arXiv:1511.08779.
25. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). Cambridge: The MIT Press.
26. Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V. F., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., & Graepel, T. (2018). Value-decomposition networks for cooperative multi-agent learning based on team reward. In E. André, S. Koenig, M. Dastani, & G. Sukthankar (Eds.) *Proceedings of the 17th international conference on autonomous agents and multiagent systems, AAMAS 2018, Stockholm, Sweden, July 10–15, 2018*, (pp. 2085–2087).

27. Rashid, T., Samvelyan, M., de Witt, C. S., Farquhar, G., Foerster, J. N., & Whiteson, S. (2018). QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In J. G. Dy & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10–15, 2018. Proceedings of machine learning research* (Vol. 80, pp. 4292–4301).
28. Son, K., Kim, D., Kang, W. J., Hostallero, D., & Yi, Y. (2019). QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning, ICML 2019, 9–15 June 2019, Long Beach, California, USA. Proceedings of machine learning research* (Vol. 97, pp. 5887–5896).
29. Wang, Y., Han, B., Wang, T., Dong, H., & Zhang, C. (2021). DOP: Off-policy multi-agent decomposed policy gradients. In *9th international conference on learning representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021*.
30. Zhang, S. Q., Zhang, Q., & Lin, J. (2019). Efficient communication in multi-agent reinforcement learning via variance based control. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32 (NeurIPS)* (pp. 3230–3239). Berlin: Springer.
31. Zhang, S. Q., Zhang, Q., & Lin, J. (2020). Succinct and robust multi-agent communication with temporal message control. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems 33 (NIPS)*.
32. Yuan, L., Wang, J., Zhang, F., Wang, C., Zhang, Z., Yu, Y., & Zhang, C. (2022). Multi-agent incentive communication via decentralized teammate modeling. In *Thirty-sixth AAAI conference on artificial intelligence (AAAI-22)*.
33. Wang, J., Ren, Z., Liu, T., Yu, Y., & Zhang, C. (2021). QPLEX: duplex dueling multi-agent q-learning. In *9th international conference on learning representations, ICLR 2021, virtual event, Austria, May 3–7, 2021*.
34. Konda, V. R., & Tsitsiklis, J. N. (1999). Actor-critic algorithms. In S. A. Solla, T. K. Leen, & K. Müller (Eds.) *Advances in neural information processing systems 12, NIPS conference* (pp. 1008–1014).
35. Schulman, J., Moritz, P., Levine, S., Jordan, M. I., & Abbeel, P. (2016). High-dimensional continuous control using generalized advantage estimation. In Y. Bengio, & Y. LeCun (Eds.) *4th International conference on learning representations, ICLR 2016, San Juan, Puerto Rico, May 2–4, 2016, Conference track proceedings*.
36. Oroojlooyjadid, A., & Hajinezhad, D. (2019). A review of cooperative multi-agent deep reinforcement learning. *CoRR*arXiv:1908.03963.
37. Papoudakis, G., Christianos, F., Schäfer, L., & Albrecht, S. V. (2020). Comparative evaluation of cooperative multi-agent deep reinforcement learning algorithms. arXiv:2006.07869.
38. Jiang, J., & Lu, Z. (2018). Learning attentional communication for multi-agent cooperation. In *Advances in neural information processing systems 31 (NIPS)*, (pp. 7265–7275).
39. Malysheva, A., Sung, T. T. K., Sohn, C., Kudenko, D., & Shpilman, A. (2018). Deep multi-agent reinforcement learning with relevance graphs. *CoRR*arXiv:1811.12557.
40. Kilinc, O., & Montana, G. (2018). Multi-agent deep reinforcement learning with extremely noisy observations. *CoRR*arXiv:1812.00922.
41. Pesce, E., & Montana, G. (2020). Improving coordination in small-scale multi-agent deep reinforcement learning through memory-driven communication. *Machine Learning, 109*(9–10), 1727–1747.
42. Kim, D., Moon, S., Hostallero, D., Kang, W. J., Lee, T., Son, K., & Yi, Y. (2019). Learning to schedule communication in multi-agent reinforcement learning. In *7th international conference on learning representations (ICLR)*.
43. Cao, K., Lazaridou, A., Lanctot, M., Leibo, J. Z., Tuyls, K., & Clark, S. (2018). Emergent communication through negotiation. In *6th international conference on learning representations, ICLR 2018, Vancouver, BC, Canada, April 30–May 3, 2018, conference track proceedings*.
44. Lowe, R., Foerster, J. N., Boureau, Y., Pineau, J., & Dauphin, Y. N. (2019). On the pitfalls of measuring emergent communication. In E. Elkind, M. Veloso, N. Agmon, & M. E. Taylor (Eds.) *Proceedings of the 18th international conference on autonomous agents and multiagent systems, AAMAS'19, Montreal, QC, Canada, May 13–17, 2019*, (pp. 693–701).
45. Bullard, K., Kiela, D., Pineau, J., & Foerster, J. N. (2021). Quasi-equivalence discovery for zero-shot emergent communication. *CoRR*arXiv:2103.08067.
46. Noukhovitch, M., LaCroix, T., Lazaridou, A., & Courville, A. C. (2021). Emergent communication under competition. In F. Dignum, A. Lomuscio, U. Endriss, & A. Nowé (eds.) *AAMAS'21: 20th international conference on autonomous agents and multiagent systems, virtual event, United Kingdom, May 3-7, 2021*, (pp. 974–982).

47. Sukhbaatar, S., Szlam, A., & Fergus, R. (2016). Learning multiagent communication with backpropagation. In *Advances in neural information processing systems 29 (NIPS)*, (pp. 2244–2252).
48. Singh, A., Jain, T., & Sukhbaatar, S. (2019). Learning when to communicate at scale in multiagent cooperative and competitive tasks. In *7th international conference on learning representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019*.
49. Peng, P., Yuan, Q., Wen, Y., Yang, Y., Tang, Z., Long, H., & Wang, J. (2017). Multiagent bidirectionally-coordinated nets for learning to play Starcraft combat games. *CoRR*arXiv:1703.10069.
50. Farrell, J., & Rabin, M. (1996). Cheap talk. *Journal of Economic Perspectives, 10*(3), 103–118.
51. Seo, H., Park, J., Bennis, M., & Debbah, M. (2021). Semantics-native communication with contextual reasoning. *CoRR*arXiv:2108.05681.
52. Taniguchi, T., Yoshida, Y., Taniguchi, A., & Hagiwara, Y. (2022). Emergent communication through metropolis-hastings naming game with deep generative models. *CoRR*arXiv:2205.12392. *https://doi.org/10.48550/arXiv.2205.12392*.
53. Chaabouni, R., Strub, F., Altché, F., Tarassov, E., Tallec, C., Davoodi, E., Mathewson, K. W., Tieleman, O., Lazaridou, A., & Piot, B. (2022). Emergent communication at scale. In *The tenth international conference on learning representations, ICLR 2022, Virtual Event, April 25–29, 2022*.
54. Chaabouni, R., Kharitonov, E., Bouchacourt, D., Dupoux, E., & Baroni, M. (2020). Compositionality and generalization in emergent languages. In D. Jurafsky, J. Chai, N. Schluter, & J. R. Tetreault (Eds.) *Proceedings of the 58th annual meeting of the association for computational linguistics, ACL 2020, Online, July 5–10, 2020*, (pp. 4427–4442).
55. Resnick, C., Gupta, A., Foerster, J. N., Dai, A. M., & Cho, K. (2020). Capacity, bandwidth, and compositionality in emergent language learning. In A. E. F. Seghrouchni, G. Sukthankar, B. An, & N. Yorke-Smith (Eds.), *Proceedings of the 19th international conference on autonomous agents and multiagent systems, AAMAS'20, Auckland, New Zealand, May 9–13, 2020* (pp. 1125–1133).
56. Chaabouni, R., Kharitonov, E., Dupoux, E., & Baroni, M. (2019). Anti-efficient encoding in emergent communication. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada* (pp. 6290–6300).
57. Havrylov, S., & Titov, I. (2017). Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems 30: Annual conference on neural information processing systems 2017, December 4–9, 2017, Long Beach, CA, USA* (pp. 2149–2159).
58. Cowen-Rivers, A. I., & Naradowsky, J. (2020). Emergent communication with world models. *CoRR*arXiv:2002.09604.
59. Kajic, I., Aygün, E., & Precup, D. (2020). Learning to cooperate: Emergent communication in multiagent navigation. In S. Denison, M. Mack, Y. Xu, & B. C. Armstrong (Eds.), *Proceedings of the 42th annual meeting of the cognitive science society—Developing a mind: Learning in humans, animals, and machines, CogSci 2020, Virtual, July 29–August 1, 2020*.
60. Mordatch, I., & Abbeel, P. (2018). Emergence of grounded compositional language in multi-agent populations. In S. A. McIlraith & K. Q. Weinberger (Eds.), *Proceedings of the thirty-second AAAI conference on artificial intelligence, (AAAI-18), the 30th innovative applications of artificial intelligence (IAAI-18), and the 8th AAAI symposium on educational advances in artificial intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2–7, 2018* (pp. 1495–1502).
61. Jaques, N., Lazaridou, A., Hughes, E., Gülçehre, Ç., Ortega, P. A., Strouse, D., Leibo, J. Z., & de Freitas, N. (2019). Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning, ICML 2019, 9–15 June 2019, Long Beach, California, USA. Proceedings of machine learning research* (Vol. 97, pp. 3040–3049).
62. Eccles, T., Bachrach, Y., Lever, G., Lazaridou, A., & Graepel, T. (2019). Biases for emergent communication in multi-agent reinforcement learning. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada* (pp. 13111–13121).
63. Tucker, M., Li, H., Agrawal, S., Hughes, D., Sycara, K. P., Lewis, M., & Shah, J. A. (2021). Emergent discrete communication in semantic spaces. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems 34: Annual conference on neural information processing systems 2021, NeurIPS 2021, December 6–14, 2021, Virtual* (pp. 10574–10586).

64. Lin, T., Huh, J., Stauffer, C., Lim, S., & Isola, P. (2021). Learning to ground multi-agent communication with autoencoders. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems 34: Annual conference on neural information processing systems 2021, NeurIPS 2021, December 6–14, 2021, Virtual* (pp. 15230–15242).
65. Wang, R., He, X., Yu, R., Qiu, W., An, B., & Rabinovich, Z. (2020). Learning efficient multi-agent communication: An information bottleneck approach. In *Proceedings of the 37th international conference on machine learning (ICML). Proceedings of machine learning research* (Vol. 119, pp. 9908–9918).
66. Xue, W., Qiu, W., An, B., Rabinovich, Z., Obraztsova, S., & Yeo, C. K. (2021). Mis-spoke or mislead: Achieving robustness in multi-agent communicative reinforcement learning. *CORR*arXiv:2108.03803.
67. Nguyen, T. T., Nguyen, N. D., & Nahavandi, S. (2018). Deep reinforcement learning for multi-agent systems: A review of challenges, solutions and applications. *CoRR*arXiv:1812.11794.
68. Zhang, K., Yang, Z., & Basar, T. (2019). Multi-agent reinforcement learning: A selective overview of theories and algorithms. *CoRR*arXiv:1911.10635.
69. Wong, A., Bäck, T., Kononova, A. V., & Plaat, A. (2021). Multiagent deep reinforcement learning: Challenges and directions towards human-like approaches. *CoRR*arXiv:2106.15691.
70. Zaïem, M. S., & Bennequin, E. (2019). Learning to communicate in multi-agent reinforcement learning : A review. *CoRR*arXiv:1911.05438.
71. Shoham, Y., & Leyton-Brown, K. (2009). *Multiagent Systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge: Cambridge University Press.
72. Foerster, J. N., Assael, Y. M., de Freitas, N., & Whiteson, S. (2016). Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems* 29 (NIPS), pp. 2137–2145.
73. Agarwal, A., Kumar, S., Sycara, K. P., & Lewis, M. (2020). Learning transferable cooperative behavior in multi-agent teams. In *Proceedings of the 19th international conference on autonomous agents and multiagent systems (AAMAS)*, pp. 1741–1743.
74. Wang, T., Wang, J., Zheng, C., & Zhang, C. (2020). Learning nearly decomposable value functions via communication minimization. In *8th international conference on learning representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*.
75. Mao, H., Zhang, Z., Xiao, Z., Gong, Z., & Ni, Y. (2020). Learning agent communication under limited bandwidth by message pruning. In *The thirty-fourth AAAI conference on artificial intelligence* (pp. 5142–5149).
76. Sheng, J., Wang, X., Jin, B., Yan, J., Li, W., Chang, T., Wang, J., & Zha, H. (2020). Learning structured communication for multi-agent reinforcement learning. *CoRR*arXiv:2002.04235.
77. Freed, B., Sartoretti, G., Hu, J., & Choset, H. (2020). Communication learning via backpropagation in discrete channels with unknown noise. In *The thirty-fourth AAAI conference on artificial intelligence* (pp. 7160–7168).
78. Ding, Z., Huang, T., & Lu, Z. (2020). Learning individually inferred communication for multi-agent cooperation. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems 33 (NeurIPS)*.
79. Yun, W. J., Lim, B., Jung, S., Ko, Y., Park, J., Kim, J., & Bennis, M. (2021). Attention-based reinforcement learning for real-time UAV semantic communication. *CoRR*arXiv:2105.10716.
80. Jiang, J., Dun, C., Huang, T., & Lu, Z. (2020). Graph convolutional reinforcement learning. In *8th international conference on learning representations* (*ICLR*).
81. Kim, W., Cho, M., & Sung, Y. (2019). Message-dropout: An efficient training method for multi-agent deep reinforcement learning. In *The thirty-third AAAI conference on artificial intelligence*, (pp. 6079–6086).
82. Liu, Y., Wang, W., Hu, Y., Hao, J., Chen, X., & Gao, Y. (2020). Multi-agent game abstraction via graph attention neural network. In *The thirty-fourth AAAI conference on artificial intelligence* (*AAAI*) (pp. 7211–7218).
83. Chu, T., Chinchali, S., & Katti, S. (2020). Multi-agent reinforcement learning for networked system control. In *8th international conference on learning representations* (*ICLR*).
84. Qu, C., Li, H., Liu, C., Xiong, J., Zhang, J., Chu, W., Qi, Y., & Song, L. (2020). Intention propagation for multi-agent reinforcement learning. *CoRR*arXiv:2004.08883.
85. Hu, G., Zhu, Y., Zhao, D., Zhao, M., & Hao, J. (2020). Event-triggered multi-agent reinforcement learning with communication under limited-bandwidth constraint. *CoRR*arXiv:2010.04978.
86. Freed, B., James, R., Sartoretti, G., & Choset, H. (2020). Sparse discrete communication learning for multi-agent cooperation through backpropagation. In *IEEE/RSJ international conference on intelligent robots and systems* (*IROS*) (pp. 7993–7998).

87. Kong, X., Xin, B., Liu, F., & Wang, Y. (2017). Revisiting the master–slave architecture in multi-agent deep reinforcement learning. *CoRR*arXiv:1712.07305.
88. Das, A., Gervet, T., Romoff, J., Batra, D., Parikh, D., Rabbat, M., & Pineau, J. (2019). Tarmac: Targeted multi-agent communication. In *Proceedings of the 36th international conference on machine learning* (*ICML*) (pp. 1538–1546).
89. Kim, W., Park, J., & Sung, Y. (2021). Communication in multi-agent reinforcement learning: Intention sharing. In *9th international conference on learning representations* (*ICLR*).
90. Gupta, N., Srinivasaraghavan, G., Mohalik, S. K., & Taylor, M. E. (2021). HAMMER: multi-level coordination of reinforcement learning agents via learned messaging. *CoRR*arXiv:2102.00824.
91. Niu, Y., Paleja, R. R., & Gombolay, M. C. (2021). Multi-agent graph-attention communication and teaming. In *20th International Conference on Autonomous Agents and Multiagent Systems* (*AAMAS*) (pp. 964–973).
92. Du, Y., Liu, B., Moens, V., Liu, Z., Ren, Z., Wang, J., Chen, X., & Zhang, H. (2021). Learning correlated communication topology in multi-agent reinforcement learning. In *20th international conference on autonomous agents and multiagent systems* (*AAMAS*) (pp. 456–464).
93. Wang, Y., & Sartoretti, G. (2022). FCMNET: Full communication memory net for team-level cooperation in multi-agent systems. *CoRR*arXiv:2201.11994.
94. Busoniu, L., Babuska, R., & Schutter, B. D. (2006). Multi-agent reinforcement learning: A survey. In *Ninth international conference on control, automation, robotics and vision* (*ICARCV*) (pp. 1–6).
95. Synnaeve, G., Nardelli, N., Auvolat, A., Chintala, S., Lacroix, T., Lin, Z., Richoux, F., & Usunier, N. (2016). Torchcraft: A library for machine learning research on real-time strategy games. *CoRR*arXiv:1611.00625.
96. Vinyals, O., Ewalds, T., Bartunov, S., Georgiev, P., Vezhnevets, A. S., Yeo, M., Makhzani, A., Küttler, H., Agapiou, J. P., Schrittwieser, J., Quan, J., Gaffney, S., Petersen, S., Simonyan, K., Schaul, T., van Hasselt, H., Silver, D., Lillicrap, T.P., Calderone, K., Keet, P., Brunasso, A., Lawrence, D., Ekermo, A., Repp, J., & Tsing, R. (2017). Starcraft II: A new challenge for reinforcement learning. *CoRR*arXiv:1708.04782.
97. Samvelyan, M., Rashid, T., de Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G. J., Hung, C., Torr, P. H. S., Foerster, J. N., & Whiteson, S. (2019). The starcraft multi-agent challenge. In E. Elkind, M. Veloso, N. Agmon, & M. E. Taylor (Eds.), *Proceedings of the 18th international conference on autonomous agents and multiagent systems, AAMAS'19, Montreal, QC, Canada, May 13–17, 2019* (pp. 2186–2188).
98. Kurach, K., Raichuk, A., Stanczyk, P., Zajac, M., Bachem, O., Espeholt, L., Riquelme, C., Vincent, D., Michalski, M., Bousquet, O., & Gelly, S. (2020). Google research football: A novel reinforcement learning environment. In *The thirty-fourth AAAI conference on artificial intelligence* (pp. 4501–4510).
99. Matignon, L., Laurent, G. J., & Fort-Piat, N. L. (2012). Independent reinforcement learners in cooperative Markov games: A survey regarding coordination problems. *Knowledge Engineering Review, 27*(1), 1–31.
100. Brys, T., Nowé, A., Kudenko, D., & Taylor, M. E. (2014). Combining multiple correlated reward and shaping signals by measuring confidence. In C. E. Brodley & P. Stone (Eds.), *Proceedings of the twenty-eighth AAAI conference on artificial intelligence, July 27–31, 2014, Québec City, Québec, Canada* (pp. 1687–1693).
101. Mao, H., Zhang, Z., Xiao, Z., Gong, Z., & Ni, Y. (2020). Learning agent communication under limited bandwidth by message pruning. In *The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, the thirty-second innovative applications of artificial intelligence conference, IAAI 2020, the tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, New York, NY, USA, February 7–12, 2020* (pp. 5142–5149).
102. Jang, E., Gu, S., & Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *5th international conference on learning representations* (*ICLR*).
103. Kraemer, L., & Banerjee, B. (2016). Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing, 190*, 82–94.
104. Granatyr, J., Botelho, V., Lessing, O. R., Scalabrin, E. E., Barthès, J. A., & Enembreck, F. (2015). Trust and reputation models for multiagent systems. *ACM Computing Surveys, 48*(2), 27–12742.
105. Gunes, D. T. (2021). *Strategic and adaptive behaviours in trust systems*. Ph.D. thesis, University of Southampton.
106. Müller, J. P., & Fischer, K. (2014). Application impact of multi-agent systems and technologies: A survey. In O. Shehory & A. Sturm (Eds.), *Agent-oriented software engineering—reflections on architectures, methodologies, languages, and frameworks* (pp. 27–53). Berlin: Springer.

107. Herrera, M., Pírez-Hernández, M., Kumar Parlikad, A., & Izquierdo, J. (2020). Multi-agent systems and complex networks: Review and applications in systems engineering. *Processes*. https://doi.org/10.3390/pr8030312
108. Calvaresi, D., Dubovitskaya, A., Calbimonte, J., Taveter, K., & Schumacher, M. (2018). Multi-agent systems and blockchain: Results from a systematic literature review. In Y. Demazeau, B. An, J. Bajo, & A. Fernández-Caballero (Eds.), *Advances in practical applications of agents, multi-agent systems, and complexity: The PAAMS collection—16th international conference, PAAMS 2018, Toledo, Spain, June 20–22, 2018, Proceedings. Lecture Notes in Computer Science* (Vol. 10978, pp. 110–126). Berlin: Springer.
109. Papoudakis, G., Christianos, F., Schäfer, L., & Albrecht, S. V. (2021). Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. In J. Vanschoren & S. Yeung (Eds.), *Proceedings of the neural information processing systems track on datasets and benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, Virtual.*
110. Bogin, B., Geva, M., & Berant, J. (2018). Emergence of communication in an interactive world with consistent speakers. *CoRR*arXiv:1809.00549.
111. Baltrusaitis, T., Ahuja, C., & Morency, L. (2019). Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 41*(2), 423–443.
112. Poklukar, P., Vasco, M., Yin, H., Melo, F. S., Paiva, A., & Kragic, D. (2022). Geometric multimodal contrastive representation learning. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, & S. Sabato (Eds.), *A concise introduction to decentralized POMDPs* (Vol. 162, pp. 17782–17800). Berlin: Springer.
113. Seering, J., Luria, M., Kaufman, G., & Hammer, J. (2019). Beyond dyadic interactions: Considering chatbots as community members. In S. A. Brewster, G. Fitzpatrick, A. L. Cox, & V. Kostakos (Eds.), *Proceedings of the 2019 CHI conference on human factors in computing systems, CHI 2019, Glasgow, Scotland, UK, May 04–09, 2019* (p. 450).
114. Seering, J., Luria, M., Ye, C., Kaufman, G., & Hammer, J. (2020). It takes a village: Integrating an adaptive chatbot into an online gaming community. In R. Bernhaupt, F. F. Mueller, D. Verweij, J. Andres, J. McGrenere, A. Cockburn, I. Avellino, A. Goguey, P. Bjøn, S. Zhao, B. P. Samson, & R. Kocielnik (Eds.), *CHI'20: CHI conference on human factors in computing systems, Honolulu, HI, USA, April 25–30, 2020* (pp. 1–13).
115. Følstad, A., & Brandtzæg, P. B. (2017). Chatbots and the new world of HCI. *Interactions, 24*(4), 38–42.
116. Choudhury, R. R., Paul, K., & Bandyopadhyay, S. (2004). Marp: A multi-agent routing protocol for mobile wireless ad hoc networks. *Autonomous Agents and Multi-Agent Systems, 8*(1), 47–68.
117. Pinto, L., Davidson, J., Sukthankar, R., & Gupta, A. (2017). Robust adversarial reinforcement learning. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th international conference on machine learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017. Proceedings of machine learning research* (Vol. 70, pp. 2817–2826).
118. Pattanaik, A., Tang, Z., Liu, S., Bommannan, G., & Chowdhary, G. (2018). Robust deep reinforcement learning with adversarial attacks. In E. André, S. Koenig, M. Dastani, & G. Sukthankar (Eds.), *Proceedings of the 17th international conference on autonomous agents and multiagent systems, AAMAS 2018, Stockholm, Sweden, July 10–15, 2018* (pp. 2040–2042).
119. Li, S., Wu, Y., Cui, X., Dong, H., Fang, F., & Russell, S. (2019). Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. In *The thirty-third AAAI conference on artificial intelligence, AAAI 2019, the thirty-first innovative applications of artificial intelligence conference, IAAI 2019, the ninth AAAI symposium on educational advances in artificial intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27–February 1, 2019* (pp. 4213–4220).
120. Zhang, K., Sun, T., Tao, Y., Genc, S., Mallya, S., & Basar, T. (2020). Robust multi-agent reinforcement learning with model uncertainty. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems 33: Annual conference on neural information processing systems 2020, NeurIPS 2020, December 6–12, 2020, Virtual*.
121. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.
122. Foerster, J. N., Nardelli, N., Farquhar, G., Afouras, T., Torr, P. H. S., Kohli, P., & Whiteson, S. (2017). Stabilising experience replay for deep multi-agent reinforcement learning. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th international conference on machine learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017. Proceedings of machine learning research* (Vol. 70, pp. 1146–1155).