The Nucleus ***60***, *No. 2 (2023) 174-184*

https://doi.org/10.71330/thenucleus.2023.1303

www.thenucleuspak.org.pk

The Nucleus

ISSN 0029-5698 (Print)

ISSN 2306-6539 (Online)

Communication in Multi-Agent Reinforcement Learning: A Survey

Rimsha Khan*, Nageen Khan, Tauqir Ahmad

Department of Computer Science, University of Engineering and Technology, Lahore, Pakistan

ABSTRACT

Agents can use communication to coordinate their actions and achieve their goals. The agents in multi-agent reinforcement learning (MARL) have the ability to enhance their overall learning performance by acquiring communication skills. They can transmit various types of messages to either all agents or particular groups, utilizing diverse communication channels. Their study on MARL with communication (Comm-MARL) is expanding. Nonetheless, currently, there is no methodical approach to differentiate and categorize present Comm-MARL (Communication Multi-agent reinforcement learning) systems. This article surveys recent research in the Comm-MARL domain, scrutinizing diverse communication aspects that could be incorporated into MARL systems. Several dimensions are suggested to examine, establish, and contrast Comm-MARL systems. This paper presents a comprehensive review of the nine dimensions influencing communication in multi-agent collaboration. The dimensions explored include communication type, communication policy, communicated messages, message combination, inner integration, communication constraints, communication learning, training schemes, and controlled goals. By examining these dimensions, the study aims to shed light on the intricate dynamics of agent interaction in complex environments. This review emphasizes the significance of effective communication strategies in achieving common objectives among agents and highlights the importance of factors such as context awareness, adaptability, and learning from past experiences. The insights provided in this paper offer valuable guidance for enhancing collaboration and communication strategies across various multi-agent systems and applications.

Keywords: *Reinforcement Learning, Multiagent, Centralized, Decentralized, Concatenation, Observable Environment.*

1. Introduction

Multi-agent collaboration refers to the process of multiple agents working together to achieve a common goal or set of goals. Collaboration among agents involves communication, coordination, and cooperation, where agents exchange information and coordinate their actions to achieve a common goal [1]. The field of multi-agent systems has gained significance because of its numerous applications in recent years, in various domains such as robotics, self-directed vehicles, and communication networks. Multi-agent collaboration is a field of research that studies how multiple agents can work together to acquire a common objective. Agents can be anything from physical robots to software programs and they can be either cooperative or competitive. The term "collective intelligence" refers to the ability of a group of people to solve problems more effectively than any individual could. This is because when people collaborate, they can pool their knowledge, skills, and perspectives to come up with better solutions [2]. In order to be successful, agents must be able to overcome a number of challenges including:

(a) Asymmetry: Agents may have different capabilities, knowledge and goals.

(b) Non-stationary: The environment may change over time and agents must be able to adapt to these changes. **Opportunism:** Agents may try to exploit each other in order to achieve their own goals. Despite these challenges, multi-agent collaboration can be a powerful tool for solving complex problems. For example, multi-agent systems have been used to control robots in manufacturing, to coordinate traffic flow and to play games. The reinforcement learning paradigm known as "multi-agent reinforcement learning" [3] (MARL) relates to interaction with both the environment and each other, with the aim of acquiring knowledge and improving their ability to achieve their respective objectives.

MARL is a challenging problem due to the fact that agents must learn to coordinate their efforts to accomplish their goals, while also avoiding being exploited by other agents. Communication can be a powerful tool for improving the performance of MARL agents [4]. By communicating with each other, agents can share information about the environment, their goals, and their current state. This information can be used to coordinate actions, avoid collisions, and learn more quickly.

1.1 Tools and Technologies

Multi-agent collaboration can be aided by a variety of tools and technologies, including [5]:

1.1.1 Agent development environments

These environments provide developers with the tools and resources they need to create and test multi-agent systems.

1.1.2 Communication protocols

These protocols define how agents can communicate with each other.

1.1.3 Distributed databases:

These databases store information that is shared among agents.

1.1.4 Machine learning algorithms

These algorithms can be used to train agents to learn from their interactions and improve their performance over time.

1.1.5 Ontologies

These ontologies define the terms and relationships that are used by agents to communicate with each other.

*Corresponding author: rimshakhan642@gmail.com

1.1.6 Simulation environments

These environments allow developers to test multi-agent systems in a controlled environment.

1.1.7 Middleware Platforms

These platforms provide a framework for developing and deploying multi-agent systems.

1.2 Algorithms used

There are many different algorithms that can be used for multi-agent collaboration and the choice of algorithm will depend on the specific application and requirement. Some common algorithms for multi-agent collaboration include [1-6]:

1.2.1 Reinforcement learning

This algorithm can be used to enable agents to learn from their surroundings and improve their performance over time.

1.2.2 Game theory

This mathematical framework can be used to model strategic interactions among agents.

1.2.3 Distributed optimization

This class of algorithms can be used to optimize a global objective function in a decentralized manner.

1.2.4 Consensus Algorithms

This class of algorithms can be used to enable agents to agree on a common value or decision.

1.2.5 Auctions and market-based mechanisms

This class of algorithms can be used to enable agents to trade goods or services with each other.

1.2.6 Communication protocols

This class of algorithms can be used to enable agents to exchange information and coordinate their actions. Numerous real-world situations involve the interaction of multiple agents that have an impact on the shared environment. Illustrative instances are self-directed driving, sensor networks, robotics, and game playing. A possible approach to addressing these problems is MARL, in which agents use reinforcement learning (RL) approaches to grow cooperative, competitive, or hybrid cooperative and competitive behaviors.

Partial observability is a key presumption in MARL [5, 6] since agents are usually scattered across the environment. Agents in MARL cannot access the status of the entire environment instead they are limited to using just their local observations. Since each agent not only interacts with a dynamic environment but also is impacted by the changing and adapting policies of other agents, MARL is also prone to the non-stationary problem [7]. Among agents, communication can stabilize learning by conveying valuable information such as observations, intentions, or experiences. This allows agents to acquire a more comprehensive understanding and coordinate their behaviors effectively [8, 9]. The present study is centered on investigating how communication can be harnessed to enhance RL agents in an environment, with a particular emphasis on learnable communication protocols. Unlike fixed communication protocols that are predetermined, learnable communication protocols are the ones that agents can acquire through learning, rather than being provided with them beforehand.

This approach aligns with recent research that highlights the importance of dynamic and adaptable communication in MARL, involving the acquisition of knowledge on when, how, and what to communicate. To achieve this, advanced deep reinforcement learning techniques have been employed [10, 11].

Despite several surveys, there is an urgent need for a systematic and organized technique to distinguish and classify Comm-MARL systems according to research on MARL enhanced with communication (Comm-MARL) [12, 13]. The design and deployment of MARL systems would be made easier by the creation of such a methodology. Think about the scenario when our intention is to build a new Comm-MARL system for a specific task. The system can be characterized using various aspects.

When developing a Comm-MARL system, the following aspects should be considered:

i. Learning to determine with whom to communicate
ii. Learning to identify when communication is necessary
iii. Learning to select the relevant information to communicate
iv. Learning to integrate and combine received information
v. Learning to determine the learning goals that can be achieved through communication

By addressing these aspects, a Comm-MARL system can be developed that maximizes the effectiveness of communication between agents and enhances the overall learning performance. We propose a multidimensional structure consisting of 9 dimensions. By thoroughly analyzing and comparing these dimensions, we aim to provide insight into the creation and design of MARL systems with communication components. Recent Comm-MARL systems may be mapped into this framework to help us better grasp the state of the art and identify important directions for future system designs.

In Section 2 of our survey, we summarize recent advancements in Comm-MARL systems, highlighting the need for a structural methodology. In Section 3, we proposed recent works categorized according to each dimension. Finally, in Section 4, conclusions based on the proposed dimensions are presented.

2. Literature Review

The seminal works of CommNet, DIAL, and RIAL have enabled deep RL agents to learn to communicate with partial observations in cooperative games. In CommNet, local observations are processed by a shared neural network and

each agent's decisions are influenced by a mean vector of messages (hidden layers) from other agents. Similarly, DIAL uses a shared network and allows agents to learn to selectively attend to certain aspects of their observations and the received messages. Reinforced Inter-Agent Learning (RIAL) introduces a centralized critic that estimates the joint value of actions taken by all agents, which is then used to guide communication among agents. These works have demonstrated the effectiveness of communication in enhancing the functionality of RL agents in multi-agent environments. Three key works in the field of multi-agent reinforcement learning are the CommNet, DIAL, and RIAL. CommNet allows agents to process their local observations using a shared neural network and their decision depends on observations. RIAL and DIAL, on the other hand, allow each agent to learn to exchange a binary or real value message that is appropriate for brief exchanges of information. These works have paved the way for many recent works in the field, which follow the end-to-end training paradigm of integrating the learning of communication and environment policy [14, 15].

Recent works [15, 16] have addressed the limitations of fully connected communication and have explored more efficient and effective ways for agents to communicate with each other. Some of these works have introduced communication constraints, such as limited bandwidth or range, to more closely mimic real-world communication scenarios. Additionally, some works have proposed communication structures that are learned or adaptive, rather than predefined, allowing agents to better tailor their communication to the specific task at hand.

Earlier approaches used a mechanism that involved a gate for each agent to determine if they should send their messages or not. ATOC [16] introduces a communication method where only agents in a specific observable area are involved. Within this group, a bi-LSTM algorithm combines the messages from each agent and sends them back to the members. IC3Net [17], an extension of CommNet deterministically decides whether to send messages to all or none. Moreover, IC3Net assigns individualized rewards to each agent instead of globally shared rewards, leading to more diverse behaviors in competitive or mixed environments. ETCNet [18] also employs a gate mechanism, but it regularizes the overall probability. In I2C [19], the causal effect of whether to communicate with others in peer to peer [20] manner is measured.

The gate mechanism allows communication alternative approaches to prioritize communication chances in a more explicit and global manner. For example, a certain number of agents are chosen by SchedNet [20] and assimilated to distribute messages. A shared graph is assimilated by GA-COMM [21], MAGIC [22], and FlowComm [23] to decide if and with whom agents should interact. GA-Comm utilizes an attention mechanism to construct an undirected communication graph, allowing pairs of agents to communicate with each other. On the other hand, MAGIC and FlowComm create a directed graph among agents, allowing for more fine-grained control over communication, with connected agents being able to communicate unilaterally or bilaterally.

Certain works adopt preexisting relationships to convey information and comprehend the messages' meaning. The agent-entity graph [24] establishes links between agents by using a pre-trained graph. Then, linked agents exchange each other encoding. Based on networked multi-agent systems (NMAS) where dispersed agents are connected, network communication [25, 26] is based on the sparse, predefined communication network. In this approach, agents exchange explicit messages during both training and execution in an NMAS.

To decide on the content of messages, many works make use of local information. In some cases, this includes individual observations [27-29]while in others it includes intended actions or plans [30, 31]. To prevent the loss of information, received messages are often concatenated [32-34]. In addition, agents may attach signatures to their messages to indicate their importance.

TarMAC and IMMAC use different methods to assign weights to received messages. While IMMAC employs softmax, TarMAC utilizes an attention mechanism. Similarly, GAComm decides whether to communicate and to determine the importance of agents. In addition, GAComm incorporates a graph neural network (GNN) to compile messages. However, a neural network can also learn implicitly about the importance of messages. Bi-LSTM layers are used in another method called BiCNet to link policy and value functions. Agents can exchange messages and learn from each other's memory states in a collaborative multi-agent reinforcement learning (Comm-MARL) system. However, such systems need to address practical constraints such as costly communication and stochastic environments.

To minimize communication overhead and contention, SchedNet [1, 35] selects to send messages to a shared communication channel. TMC [35] disallows agents from broadcasting similar messages and stores received messages in a buffer to compensate for missing messages.

Gated-ACML [36 , 37] introduces a two-step approach that aggressively removes messages. Similar to ATOC and IC3Net, the initial phase is studying a gate mechanism to determine whether to send messages or not. The second phase involves the assumption of a centralized message coordinator by Gated-ACML, who organizes the messages and distributes them to each agent. This strategy lessens communication overhead, as each agent theoretically only needs to communicate with the coordinator.

2.1 MARL system

Two methods [37] for coping with communication constraints in ETCNet use MARL. and variable-length coding. ETCNet computes an upper limit on the likelihood that agents can communicate at each time step and optimizes policies accordingly. In contrast, variable-length coding

permits agents to control the quantity of bits they send out at any one moment. While both methods are effective in enhancing the performance of MARL agents, ETCNet is computationally more demanding than variable-length coding.

3. Proposed dimensions

We aim to provide a systematic and structured approach to design Comm-MARL. We suggest that there are nine dimensions that may be used to describe Comm-MARL systems. Many aspects and the target issues are shown in Table 1. In the following subsections, recent works in Comm-MARL have been summarized and classified based on the proposed dimensions.

Table 1: Dimensions and their targeted problems

Dimensions	Targeted problems
Communication type	Which type of agents to communicate with?
Communication policy	When and how to build communication links among agents?
Communicated messages	Which piece of information to share?
Message combination	How to combine received messages?
Inner integration	How to integrate combined messages into learning models?
Communication constraints	How to fulfill realistic requirements?
Communication learning	How to train and improve communication?
Training schemes	How to utilize collected experience from agents?
Controlled goals	What kind of behaviors are desired to emerge with communication

3.1 Communicates Type

In Comm-MARL systems, the communicate type dimension refers to the classification of agents based on whether they communicate with each other directly or not. In the literature, this dimension has been classified into different categories.

Table 2: Category of communication type

Types	Sub types	Methods
Agents in the MAS	Nearby AGENTS	DGN[38];MAG-NET-SA-GS-MG[46]; AGENT-ENTITY GRAPH[29]; LSC[45]; NEURCOMM[32];IP[33];FLOWCOMM[14]; GAXNET[44]
	Other agents	DIAL[11];RIAL[11];COMMNET[12];BICNET[36];TARMAC[20];MADDPGM[47];IC3NET[24];SCHEDNET[25];DCCMD[48];VBC[39];DIFFDISCRETE[49];12C[28];IS[34];ETCNET[27];VARIABLE LENGTH CODING[43]; TMC [40]
PROXY	====	MAGIC[13];GA-COMM[22];MS-MARL-GCM[50]; IMAC[21]; ATOC[23]; MD-MADDPG[37]; GATED-ACML[26]; HAMMER[51];

3.2 Agents in MAS

In this category, some agents may not be able to communicate with all other agents in the MAS. Therefore, two types of agents are distinguished. Type 1 agent can communicate with all other agents in the MAS and Type 2 agents can only communicate with a subset of agents in the MAS. In some MARL systems, agents can only communicate with nearby agents, which can be defined in different ways such as agents that may be observed or nearby agents.

3.2.1 Neighboring agents on a graph:

In this communication type, agents are connected to each other on a graph and they can only communicate with their neighboring agents. This can be defined in a few ways, such as:

- Observable agents: Agents that can observe each other
- Agents within a certain distance: Agents that are within a certain distance of each other
- Neighboring agents: Agents that are connected to each other on a graph

For instance, GAXNet [38] allows agents that are observable to each other to communicate. DGN [39] limits communication to the three closest neighbors of each agent. The agent-entity graph [40] measures the distance between agents and enables communication between any two agents that are close to one another. Agents within a cluster radius can choose whether to become leader agents using LSC [41].

Both the networked multi-agent systems NeurComm [42] and IP [43] employ an established graph structure among their agents. As a result, communication between agents is limited to those that are related to one another on the graph. Similar to this, MAGNet-SA-GS-MG [44] likewise limits agent communication by using a pre-trained graph.

3.2.2 Other (Learning) Agents:

In certain MARL setups, communication between agents can occur without any proximity constraints. An example is IC3Net [45], where learning agents can communicate with their opponents, even if they have fixed policies. The results of experiments show that opponents eventually learn to communicate to avoid exploitation. Another way to facilitate communication in MARL is to use a proxy agent. This agent acts as a central point for communication among other agents but has no direct impact on the environment. Its role is to coordinate and transform messages between agents.

Various types of proxies can be used to facilitate communication among agents in a Comm-MARL system [44-46]. A scheduler that collects encoded data from all agents and communicates with each one individually, a message coordinator that allows agents to decide whether to communicate with each other, and a system that combines messages from agents depending on their weights. Table 2 provides an overview of recent works on communication type.

Agent 1 can communicate directly with nearby agents 3 and 4, but also with agent 2 via a central proxy. The proxy coordinates and transforms messages to allow agent 1 and agent 2 to communicate with each other, despite not being nearby agents.

3.3 Communication policy

A communication policy is a set of rules that govern how agents communicate with each other. Predefined communication policies are fixed and do not change during learning. For example, a predefined communication policy might allow all agents to communicate with each other, or it might only allow agents to communicate with their neighbors. Learned communication policies are updated during learning. For instance, an agent might learn to communicate with other agents that have been helpful in the past, or it might learn to communicate with agents that have similar goals. There are four main categories of communication policies: [46-49, 63]

Full communication: All agents are permitted to speak with one another.

Partial structure: Agents can communicate with a subset of other agents, based on a predefined rule.

Individual control: Each agent learns its own communication policy.

Global control: A central agent learns a communication policy for all agents.

Full communication is the simplest communication policy. It is permitted for all agents to converse with one another regardless of their location or goals. This policy is easy to implement, but it can be inefficient if agents are not communicating with the agents that are most relevant to them. Partial structure is a more complex communication policy. Only a selected group of other agents are permitted to communicate based on a predefined rule. This rule can be based on the agents' location, their goals or any other relevant factor. Partial structure can improve efficiency by reducing the number of unnecessary messages.

Individual control is a more flexible communication policy. Each agent learns its own communication policy. This policy can be more efficient than partial structure because agents can learn to communicate with the agents that are most relevant to them. However, individual control can be more difficult to implement because each agent needs to learn its own policy. Global control is the most complex communication policy. A central agent learns a communication policy for all agents. This policy can be the most efficient because the central agent can take into account the communication needs of all agents. However, global control can be the most difficult to implement because the central agent needs to possess a thorough awareness of the environment and the objectives of the agents.

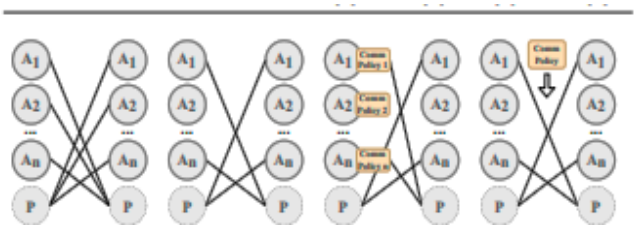

Fig. 1: **C**ommunication standards with agents: [46-49, 63]

The choice of communication policy depends on the specific application. If efficiency is the most important factor, then full communication or partial structure might be the best choice. If flexibility is the most important factor, then individual control might be the best choice. If accuracy is the main significant factor, then global control might be the best choice**.**

3.4 Communicated Messages

Once agents have established communication links, they need to decide what information to share. Partial observability means that each agent only has access to a limited view of the environment, so sharing local observations can be helpful for coordination. Agents can also share historical experiences, desired courses of action, or upcoming plans to produce more enlightening communications.

Recent research in this field varies depending on whether further information is simulated and encoded and can be classified into two categories:

3.4.1 Non-Future-Aware Communication:

In this category, agents do not consider future information when deciding what to communicate. They only share local observations, historical experiences, or intended actions.

3.4.2 Future-Aware Communication:

In this category, agents consider future information when deciding what to communicate. They may simulate future outcomes, encode future plans, or use other methods to generate more informative messages.

The choice of communication strategy depends on the specific application. If efficiency and accuracy are the most important factors, then non-future-aware communication might be the best choice.

3.5 Message Combination [47]

To process received messages, most existing Comm-MARL works typically handle them as a single entity. Message combination involves merging multiple received messages into a single message that can be used in an agent's internal model. In the presence of a proxy, agents typically receive a coordinated and combined message from the proxy, which takes care of the message combination process as discussed in the communicated messages dimension. In the absence of a proxy, agents need to individually determine how to combine multiple messages. As communicated messages capture the senders' interpretation of some messages may be more important than others depending on the situation or the learning process. For example, a message from an agent who has a lot of experience in the environment may be more valuable than a message from an agent who is new to the environment. There are a few different approaches to message combination. One approach is to simply average the values of the messages. Another approach is to use a weighted average, where the weights are determined by the senders' expertise or other factors. A third approach is to use

a more complex algorithm, such as a neural network to combine the messages. The choice of message combination approach depends on the specific application. If efficiency is a highly valued factor, then a simple approach, such as: averaging might be the best choice. If accuracy is the most important factor, then a more complex approach, such as: using a neural network might be the best choice. There are three [48-49] main approaches to message combination: concatenation, equal weighting, and unequal weighting. Their discussion is presented below:

3.5.1 Concatenation:

In this approach, the messages are simply concatenated together. This means that no preference is introduced and all messages are treated equally. Concatenation can be a good approach if the messages are relatively short and if it is important to preserve all of the information in the messages.

3.5.2 Equal Weighting:

In this approach, the messages are weighted equally. This means that each message is given the same weight when it is combined with the other messages. Equal weighting can be a good approach if the messages are all of similar quality and if it is important to keep the message combination process simple.

3.5.3 Unequal Weighting:

In this approach, the messages are weighted unequally. This means that some messages are given more weight than others. Unequal weighting can be a good approach if the messages are of different quality. The weighting of the data may be done in a variety of ways. One of the methods is messages unequally. One common method is to use an attention mechanism. Attention mechanisms assign weights to the messages based on their relevance to the current task. Another common method is to use a neural network. Neural networks can learn to weight the messages automatically, based on the data.

The choice of message combination approach depends on the specific application. If efficiency is the most important factor, then concatenation might be the best choice. If accuracy is the most important factor, then unequal weighting might be the best choice.

3.6 Inner Integration

For this, the literature generally approaches the concept of inner integration. Messages are typically seen as additional observations, which can be fed as an extra input either to a value function, a policy function, or both.

3.6.1 Policy-level:

To integrate messages into a policy model, agents can use the received messages to choose the next action. This way, agents can exploit information from other agents and act in a coordinated manner rather than independently. There are various methods for learning the policy model, such as policy gradient with REINFORCE, which gains reward in episodes and trains the model at the end of each episode, and actor-critic methods, which use a Q-function as a critic model to guide the learning of a policy network as an actor model.

3.6.2 Value-level:

Messages are considered as input to a value function, also known as an action-value function, in this category. DQN-like methods are commonly used in most of the works in this category.

3.6.3 Policy-level & Value-level

Integrating messages into both a policy and value model typically involves applying actor-critic techniques. These approaches use the messages that have been sent as additional inputs for the actor and critic models, respectively. As an alternative, the messages might be used in conjunction with nearby observations to produce fresh internal states that could then be communicated to both the actor and critic models.

The choice of the inner integration approach depends on the specific application. If efficiency is the most important factor, then policy-level integration might be the best choice. If the highest priority is accuracy, then integrating messages at the policy and value levels could be the optimal option.

3.7 Communication Constraints

To address realistic challenges such as the price of communication and noisy environments, systems Comm-MARL need to consider communication constraints. Recent work in this domain can be classified into three types:

3.7.1 Limited communication rounds:

In this type of constraint, agents are only allowed to communicate a limited number of times. This can be a realistic constraint in applications where communication is expensive such as in satellite communications.

3.7.2 Noisy communication:

In this type of constraint, messages can be corrupted or lost. This can be a realistic constraint in applications where communication is unreliable such as: in wireless communications.

3.7.3 Partial observability:

In this type of constraint, agents do not have access to all environmental information. This can be a realistic constraint in applications where it is difficult or impossible at the end of each episode, and obtain complete information about both the model and actor-critic techniques that employ a Q-function.

The choice of communication constraint depends on the specific application. If efficiency is the highly valued factor, then limited communication rounds might be the best choice. If accuracy is the most important factor, then noisy communication or partial observability might be the best choice.

3.8 Limited Bandwidth

Communication constraints refer to the limitations of communication bandwidth and capacity, which can affect the performance of Comm-MARL systems in realistic settings. Recent work [40-49, 57] in this domain can be classified into three types:

The first type assumes that because of the restricted communication bandwidth and capacity, early works try to send brief messages to cut down on communication overhead. DIAL [46] and RIAL [45] propose binary and real-value messages, respectively, to communicate between two agents and alleviate limited channel capacity. SchedNet [47] selects a subset of agents to broadcast their messages according to their importance. With reduced communication cost than SchedNet, VBC [48] and TMC [49] use specified criteria to filter out extraneous traffic. A probabilistic gate unit to block messages and a message coordinator are learned via gated ACML, which requires adjusting the gate through learning compared to handcrafted thresholds. These methods all attempt to reduce communication overhead in order to enhance Comm-MARL systems' performance in low bandwidth settings.

3.8.1 Noisy Channel:

In this categorization, ambient noise may interfere with how signals are sent between agents. The DIAL method, which works with Gaussian noise, shows how adding noise to real-valued signals may change their distribution. Contrarily, discrete attempts to back-propagate derivatives using discretized real-valued messages while addressing a noisy channel. Gradients may be obtained by using the suggested approach, which mathematically equates transmitting real-value signals to the discretized signal with additive noise [50-52].

3.8.2 Shared Medium:

This division of disagreement deals with situations where messages are transmitted through a single medium. MD-MADDPG solves this problem by allowing agents to access shared memory space sequentially to avoid contention. SchedNet selects agents who place a high priority on spreading their message, reducing the likelihood of contention.

3.8.3 Communication Learning:

Communication Learning deals with updating and adjusting a communication protocol through acquiring knowledge about a communication strategy and message. The education process can utilize defined feedback in the form of rewards, which can be enhanced in addition to learning the environment policy by a reinforcement learning process. To give richer and denser feedback, gradients can also be back-propagated from one agent to another. However, if agents learn discrete sending behaviors, this approach may be impractical because it necessitates differential messages and communication behaviors. On the basis of how communication feedback is used, recent studies in this dimension are grouped.

3.8.4 Reinforced:

In this category, the communication protocol is trained using reinforcement learning methods. RIAL and HAMMER concentrate on teaching the message's content in its entirety without considering whether to communicate. Other works consider both the learning of message content and the decision of whether to communicate. In addition, most works employ a centralized critic to provide feedback to all agents, while some works use a decentralized critic to provide feedback to individual agents.

3.8.5 Policy Gradient:

In this category, the communication policy is learned using policy gradient methods. This approach is more efficient than reinforcement learning but it requires the communication policy to be differentiable.

3.8.6 Backpropagation:

In this category, gradients are back-propagated through the communication policy. This approach is more accurate than policy gradient, but it requires the communication policy to be differentiable and the messages to be discretized.

3.9 Imitation Learning

In this category, an expert communication policy is observed and imitated by the agents. This approach is simple to implement but it requires an expert communication policy to be available. The choice of communication learning approach depends on the specific application. If efficiency is the most important factor, then reinforced learning might be the best choice. If accuracy is the most important factor, then back-propagation or imitation learning might be the best choice.

4. Training Scheme

It deals with how to leverage the accumulated experience from the agents in a Comm-MARL system including observations, actions, incentives, and messages. One approach [53] is to train each agent's model in a decentralized manner using its own experience. Another approach [53-55] is to train the agents centrally resulting in a single model to control. However, both have their drawbacks. Decentralized learning needs to handle a non - non-non-stationary environment due to agents that change and adapt. On the other hand, centralized learning faces a stationary environment while also struggling with a sizable common policy area that can be challenging to search.

A common approach [43-47, 55, 60-63] that balances the benefits of centralized and decentralized training is the CTDE technique, where agents get centralized training and decentralized execution and learn their local policies while being guided by central information. Recent works [12, 13]on training schemes can be classified based on how agents' experiences are utilized.

(a) Centralized Learning: Recent works in this area do not assume the presence of a central controller during the

execution of tasks, where experiences are collected into a central unit for learning to control all agents.

(b) Decentralized Learning: Each agent in this category develops their own policy independently. This approach is simple to implement, but it can be slow to converge.

(c) Centralized-Decentralized Learning: In this category, agents first learn a centralized policy, which is then decomposed into decentralized policies. This approach can be more efficient than decentralized learning, but it can be more difficult to implement.

(d) Federated Learning: In this category, agents learn their own policies locally and then exchange information with each other. This approach can be more efficient than centralized learning, but it can be more difficult to implement. The choice of training scheme depends on the specific application. If efficiency is the highly valued factor, then decentralized learning might be the best choice. If accuracy is the most important factor, then centralized learning or federated learning might be the best choice.

4.1 Decentralized Learning

Experiences are collected on an individual basis and agents are trained independently in this category. While this approach is straightforward to implement, it may converge slowly.

4.2 CTDE

We can further classify recent works in CTDE based on how they utilize experiences from all agents for optimization. Parameter sharing is also crucial for improving data efficiency by allowing a single set of parameters and information might be exchanged across agents as a Q-function or a policy. Agents may nevertheless behave differently since they get unique observations at each time step. These insights allow us to distinguish between the most recent researches in the following groupings [56-59].

4.3 Personalized (Policy) Parameters

Local policies in this scenario possess distinct sets of parameters and a central unit is responsible for accumulating all experiences to offer universal guidance and information, including gradients. To train the complete system, we can utilize techniques like the policy gradient actor-critic-based techniques or an algorithm (like REINFORCE).

4.4 Parameter Sharing

Typically, in this situation, DQN-like algorithms, actor-critic-based techniques, and policy gradients with REINFORCE are used. A local Q-function will learn to take into account all experiences if a DQN-like method is used or a separate global Q-function will be used to direct the learning. A shared actor (i.e., policy model) is trained to include all observation-action pairings when an actor-critic-based technique is utilized and it gets gradient instruction from a central critic [12]. Now here is the summarization of the whole discussion in a comparison table named Table 3. In Table 3 we have discussed all 9 dimensions of the communication used in MARL system and what are their further methods of implementation

Table 3: Comparative Analysis of Dimensions of Communication in Multi-Agent Reinforcement Learning

Dimension	Description	Examples
Communication type	The way in which agents communicate with each other. direct or indirect.	*__Direct communication__: Agents send and receive messages. *__Indirect communication__: Agents learn to coordinate their actions without explicitly communicating.
Communication policy	The rules that govern how agents communicate.	*__Full communication__ *__Partial structure__ *__Individual control__ *__Global Control__
Communicated messages	The information that agents share with each other.	*__Non-Future aware communication.__ *__Future aware communication.__
Message combination	The way in which agents combine the messages they receive.	*__Concatenation:__ Agents concatenate the messages they receive into a single message. *__Equal weighing:__ The messages are weighted equally. *__Unequal message weighing:__ The messages are not weighted equally.
Inner integration	The way in which agents incorporate the messages they receive into their decision-making process.	*__Policy level:__ To integrate messages into a policy model, agents can use the received messages to choose the next action. *__Value level__: Messages are considered as input to a value function. *__Policy-level & Value-level__ Integrating messages into both a policy and value model typically involves applying actor-critic techniques.
Communication constraints	The limitations on how agents can communicate.	*__Limited communication rounds__: In this type of constraint, agents are only allowed to communicate a limited number of times. * __Noisy communication:__ In this type of constraint, messages can be corrupted or lost. * __Partial observability:__ In this type of constraint, Agents do not have access to all environmental information.
Communication learning	The way in which agents learn to communicate.	*__Supervised learning:__ Agents are given a reward for sending messages that lead to good outcomes. *__Reinforcement learning__: Agents learn to communicate by trial and error.
Training schemes	The way in which agents are trained to communicate.	*__Independent training__: Agents are trained separately. *__Cooperative training:__ Agents are trained together.
Controlled goals	The goals that agents are trying to achieve through communication.	*__Coordination__: Agents are trying to coordinate their actions to achieve a common goal. *__Competition:__ Agents are trying to compete with each other to achieve a better outcome.

4.5 Concurrent

Agents are allowed to make a backup of all experiences by observing the behavior and observations of other agents.

Compared to decentralized learning, this is different. In this case, while getting guidance including global information, each local policy maintains a unique set of parameters. Actor-critic-based methods are frequently employed in concurrent CTDE, where each agent has its central critic to guide its local policy.

4.6 Controlled Goal

It is expected to achieve its aim by defining a reward configuration. The behavior correlates to different reward setups and learning may be divided into three types: cooperative, competitive, and mixed. It is worth mentioning that some studies have examined multiple scenarios to demonstrate their flexibility and scalability. Recent Research [13] in this area is divided into categories according to the behavior that is intended to emerge among learning agents with varying reward configurations.

4.7 Cooperative

In a cooperative situation, actors are compelled to interact in order to improve performance as a whole of the team. A team of agents may receive a collective reward that does not consider the contribution of each agent separately. As an alternative, agents might get regional incentives that are created to be dependent on the collective performance of their teammates, penalize collisions, or share rewards with neighbors to foster cooperation [14].

4.8 Competitive

StarCraft is a prevalent testing environment that involves multiple competitive teams. However, most studies only control one team and therefore, are not of interest to us. Based on our observations, only one study, IC3Net has examined samples having antagonistic benefits. IC3Net reveals that only when it is necessary, competing agents learn to communicate.

4.9 Mixed

In a mixed scenario, agents can exhibit either cooperative or competitive behavior, depending on the circumstances. For instance, agents may collaborate to achieve a shared objective but then compete for rewards after the objective is accomplished.

5. Conclusion

We have identified nine dimensions for analyzing and comparing various CommMARL systems, which researchers can utilize to develop their Comm-MARL system. Despite the success in this field, there are still some issues that require further examination and resolution. Most recent studies on Comm-MARL employ communication that is facilitated by the Sender-Receiver or Sender-Proxy-Receiver paradigm. This approach is helpful for learning since gradients may easily be back propagated from communicatees. Agents can, however, also ask their counterparts for specific data. Take Xuan et al. as an example. [12, 13] outline that agents can ask questions or sync their knowledge.

Secondly, as discussed, communication limitations are crucial for scenarios that necessitate low communication costs and reliable communication, necessitating further investigation and integration with practical use cases.

Thirdly, assessing the impact of a communication protocol is tough since it is hard to tell whether performance improvements are due to messages sent or environmental events. We have found two methods for learning communication in the dimension of communication learning protocol: reinforcement and differential. However, reinforcement learning requires human input to design suitable feedback for learning, while differential learning may face difficulties in determining how each agent contributes to a shared reward. Thus, it is necessary to develop more advanced and efficient methods for learning communication protocols.

Moreover, parameter sharing has become popular in recent works but it assumes homogeneous learning models. Developing Comm-MARL systems for heterogeneous agents is an area that needs further exploration. We are optimistic about the future of Comm-MARL. With the continued advancement of artificial intelligence, we can anticipate the development of more complex and sophisticated Comm-MARL systems that can tackle more challenging problems.

References

[1] C. Amato, A. Oliehoek, and C.A. T. Juan, "A Concise Introduction to Decentralized POMDPs," Springer, 2016.

[2] M.S. Zaïem, M. Etienne, and Bennequin, "Learning to communicate in multi-agent reinforcement learning," CoRR, abs/1911.05438, 2019.

[3] G. Papoudakis, F. Christianos, A. Rahman, and S.V. Albrecht, "Dealing with non-stationarity in multi-agent deep reinforcement learning," CoRR, abs/1906.04737, 2019.

[4] J.N. Foerster, G. Farquhar, T.A. fouras, N. Nardelli, and S. Whiteson, "Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence," AAAI Press, pp. 2974-2982, 2018.

[5] J.N. Foerster, Y.M. Assael, N. de Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in Advances in Neural Information Processing Systems, vol. 29, no. (NIPS), pp. 2137-2145, 2016.

[6] J. Kober, J.A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," Int. J. Robotics, vol. 32, no. (11), pp. 1238-1274, 2013.

[7] M. Vinyals, J.A. Rodríguez-Aguilar, and J. Cerquides, "A survey on sensor networks from a multiagent perspective," Comput. J., vol. 54, no. 3, pp. 455-470, 2011.

[8] P. Hernandez-Leal, B. Kartal, and M.E. Taylor, "A survey and critique of multiagent deep reinforcement learning," Autonomous Agents and Multi-Agent Systems, vol. 33, no. 6, pp. 750-797, 2019.

[9] P. Hernandez-Leal, B. Kartal, and M.E. Taylor, "A survey and critique of multiagent deep reinforcement learning," Autonomous Agents and Multi-Agent Systems, pp. 750-797, 2019.

[10] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in Advances in Neural Information Processing Systems, vol. 30, no. (NIPS), pp. 6379-6390, 2017.

[11] S. Sukhbaatar, A. Szlam, and R. Fergus, "Learning multiagent communication with backpropagation," in Advances in Neural Information Processing Systems, vol. 29, no. (NIPS), pp. 2244-2252, 2016.

[12] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent reinforcement learning for autonomous driving," CoRR, abs/1610.03295, 2016.

[13] N. Sandholm, T. Brown, and T. Tuomas, "Superhuman AI for multiplayer poker," Science, vol. 365, no. (9495), pp. 885-890, 2019.

[14] Y. Du, B. Liu, V. Moens, Z. Liu, Z. Ren, J. Wang, X. Chen, and H. Zhang, "Learning correlated communication topology in multi-agent reinforcement learning," in 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 456-464, 2021.

[15] Y. Niu, R.R. Paleja, and M.C. Gombolay, "Multi-agent graph attention communication and teaming," in 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 964-973, 2021.

[16] T.T. Nguyen, N.D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multi-agent systems: A review of challenges, solutions, and applications," CoRR, abs/1812.11794, 2018.

[17] Lee, J., Lee, M, Lee J., & Choi S. (2021). Deep reinforcement learning for multiagent collaboration arXiv preprint arXiv 210209561.

[18] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: a survey," Artificial Intelligence, pp. 1-49, 2021.

[19] A. Oroojlooyjadid and D. Hajinezhad, "A review of cooperative multiagent deep reinforcement learning," CoRR, abs/1908.03963, 2019.

[20] A. Wong, T. Bäck, A.V. Kononova, and A. Plaat, "Multiagent deep reinforcement learning: Challenges and directions towards human-like approaches," CoRR, abs/2106.15691, 2021.

[21] A. Das, T. Gervet, J. Romoff, D. Batra, D. Parikh, M. Rabbat, and J. Pineau, "Tarmac: Targeted multi-agent communication," in Proceedings of the 36th International Conference on Machine Learning (ICML), pp. 1538-1546, 2019.

[22] R. Wang, X. He, R. Yu, W. Qiu, B. An, and Z. Rabinovich, "Learning efficient multi-agent communication: An information bottleneck approach," in Proceedings of the 37th International Conference on Machine Learning (ICML), vol. 119, pp. 9908-9918, 2020.

[23] Y. Liu, W. Wang, Y. Hu, J. Hao, X. Chen, and Y. Gao, "Multi-agent game abstraction via graph attention neural network," in The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI), pp. 7211-7218, 2020.

[24] J. Jiang and Z. Lu, "Learning attentional communication for multiagent cooperation," in Advances in Neural Information Processing Systems 31 (NIPS), pp. 7265-7275, 2018.

[25] A. Singh, T. Jain, and S. Sukhbaatar, "Individualized controlled continuous communication model for multiagent cooperative and competitive tasks," in International Conference on Learning Representations (ICLR), 2019.

[26] D. Kim, S. Moon, D. Hostallero, W. J. Kang, T. Lee, K. Son, and Y. Yi, "Learning to schedule communication in multiagent reinforcement learning," in 7th International Conference on Learning Representations (ICLR), no. OpenReview.net, 2019.

[27] H. Mao, Z. Zhang, Z. Xiao, Z. Gong, and Y. Ni, "Learning agent communication under limited bandwidth by message pruning," in The Thirty-Fourth AAAI Conference on Artificial Intelligence, no. AAAI Press, pp. 5142-5149, 2020.

[28] G. Hu, Y. Zhu, D. Zhao, M. Zhao, and J. Hao, "Event-triggered multi-agent reinforcement learning with communication under limited-bandwidth constraint," CoRR, abs/2010.04978, 2020.

[29] Z. Ding, T. Huang, and Z. Lu, "Learning individually inferred communication for multi-agent cooperation," in Advances in Neural Information Processing Systems 33 (NeurIPS), 2020.

[30] A. Agarwal, S. Kumar, K. P. Sycara, and M. Lewis, "Learning transferable cooperative behavior in multi-agent teams," in International Foundation for Autonomous Agents and Multiagent Systems, pp. 1741-1743, 2020.

[31] K. Zhang, Z. Yang, and T. Basar, "Decentralized multi-agent reinforcement learning with networked agents," Frontiers Inf. Technol. Electron. Eng., vol. 22, no. 6, pp. 802-814, 2021.

[32] K. Zhang, Z. Yang, and T. Basar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," CoRR, abs/1911.10635, 2019.

[33] T. Chu, S. Chinchali, and S. Katti, "Multi-agent reinforcement learning for networked system control," in 8th International Conference on Learning Representations (ICLR), no. OpenReview.net, 2020.

[34] C. Qu, H. Li, C. Liu, J. Xiong, J. Zhang, W. Chu, Y. Qi, and L. Song, "Intention propagation for multi-agent reinforcement learning," CoRR, abs/2004.08883, 2020.

[35] W. Kim, J. Park, and Y. Sung, "Communication in multi-agent reinforcement learning: Intention sharing," in 9th International Conference on Learning Representations (ICLR), 2021.

[36] C. Sun, B. Wu, R. Wang, X. Hu, X. Yang, and C. Cong, "Intrinsic motivated multi-agent communication," in AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, 2021.

[37] P. Peng, Q. Yuan, Y. Wen, Y. Yang, Z. Tang, H. Long, and J. Wang, "Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games," CoRR, abs/1703.10069, 2017.

[38] E. Pesce and G. Montana, "Improving coordination in small-scale multi-agent deep reinforcement learning through memory-driven communication," Machine Learning, vol. 109, no. 9-10, pp. 1727-1747, 2020.

[39] J. Jiang, C. Dun, T. Huang, and Z. Lu, "Graph convolutional reinforcement learning," in 8th International Conference on Learning Representations (ICLR), no. OpenReview.net, 2020.

[40] S. Q. Zhang, Q. Zhang, and J. Lin, "Efficient communication in multi-agent reinforcement learning via variance-based control," in Advances in Neural Information Processing Systems 32 (NeurIPS), pp. 3230-3239, 2019.

[41] S. Q. Zhang, Q. Zhang, and J. Lin, "Succinct and robust multi-agent communication with temporal message control," in Advances in Neural Information Processing Systems 33 (NeurIPS), 2020.

[42] C. E. Shannon, "A mathematical theory of communication," Bell Syst. Tech. J., vol. 27, no. 3, pp. 379-423, 1948.

[43] R. L. Freeman, "Telecommunication system engineering," John Wiley & Sons, vol. 82, 2004.

[44] B. Freed, R. James, G. Sartoretti, and H. Choset, "Sparse discrete communication learning for multi-agent cooperation through backpropagation," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 7993-7998, 2020.

[45] W.J. Yun, B. Lim, S. Jung, Y.C. Ko, J. Park, J. Kim, and M. Bennis, "Attention-based reinforcement learning for real-time UAV semantic communication," CoRR, abs/2105.10716, 2021.

[46] J. Sheng, X. Wang, B. Jin, J. Yan, W. Li, T. H. Chang, J. Wang, and H. Zha, "Learning structured communication for multiagent reinforcement learning," CoRR, abs/2002.04235, 2020.

[47] A. Malysheva, T.T. K. Sung, C. Sohn, D. Kudenko, and A. Shpilman, "Deep multi-agent reinforcement learning with relevance graphs," CoRR, abs/1811.12557, 2018.

[48] O. Kilinc and G. Montana, "Multi-agent deep reinforcement learning with extremely noisy observations," CoRR, abs/1812.00922, 2018.

[49] W. Kim, M. Cho, and Y. Sung, "An efficient training method for multi-agent deep reinforcement learning," in The Thirty-Third AAAI Conference on Artificial Intelligence, pp. 6079-6086, 2019.

[50] B. Freed, G. Sartoretti, J. Hu, and H. Choset, "Communication learning via backpropagation in discrete channels with unknown noise," in The Thirty-Fourth AAAI Conference on Artificial Intelligence, pp. 7160-7168, 2020.

[51] X. Kong, B. Xin, F. Liu, and Y. Wang, "Revisiting the master-slave architecture in multi-agent deep reinforcement learning," CoRR, abs/1712.07305, 2017.

[52] N. Gupta, G. Srinivasaraghavan, S. K. Mohalik, and M. E. Taylor, "HAMMER: multi-level coordination of reinforcement learning agents via learned messaging," CoRR, abs/2102.00824, 2021.

[53] H. Mao, Z. Zhang, Z. Xiao, Z. Gong, and Y. Ni, "Learning agent communication under limited bandwidth by message pruning," in The

Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI), AAAI Press, 2020.

[54] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in 5th International Conference on Learning Representations (ICLR), OpenReview.net, 2020.

[55] L. Kraemer and B. Banerjee, "Multi-agent reinforcement learning as a rehearsal for decentralized planning," Neurocomputing, vol. 190, pp. 82-94, 2016.

[56] L. Busoniu, R. Babuska, and B. De Schutter, "Multi-agent reinforcement learning: A survey," in Ninth International Conference on Control, Automation, Robotics and Vision (ICARCV), IEEE, pp. 1-6, 2006.

[57] L. Matignon, G.J. Laurent, and N. Le Fort-Piat, "Independent reinforcement learners in cooperative Markov games," Knowledge Engineering Review, vol. 27, no. 1, pp. 1-31, 2012.

[58] B. Liu, Q. Liu, P. Stone, A. Garg, Y. Zhu, and A. Anandkumar, "Coach-player multi-agent reinforcement learning for dynamic team composition," Proceedings of Machine Learning Research, vol. 144, pp. 6860-6870, 2021.

[59] P. Xuan, V.R. Lesser, and S. Zilberstein, "Communication decisions in multi-agent cooperation: model and experiments," in Proceedings of the Fifth International Conference on Autonomous Agents, 2001.

[60] V. Hakami, V. Barghi, H. Mostafavi, S. Arefinezhad, and Z. Hakami, "A resource allocation scheme for D2D communications with unknown channel state information," Peer-to-Peer Netw. Appl., vol. 15, pp. 1189-1213, 2022.

[61] Y. Du, B. Liu, V. Moens, Z. Liu, Z. Ren, J. Wang, X. Chen, and H. Zhang, "Learning correlated communication topology in multi-agent reinforcement learning," in 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 456-464, 2021.

[62] Y. Niu, R.R. Paleja, and M.C. Gombolay, "Multi-agent graph attention communication and teaming," in 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 964-973, 2021.

[63] D. Silver et al., "Mastering the game of Go without human knowledge," Nature, vol. 550, no. 7676, pp. 354-359, 2017.