一个非常详尽的 survey，但没有关注到 多轮

A Survey of Multi-Agent Reinforcement Learning with Communication

Changxi Zhu
Utrecht University
c.zhu@uu.nl

Mehdi Dastani
Utrecht University
m.m.dastani@uu.nl

Shihan Wang
Utrecht University
s.wang2@uu.nl

ABSTRACT

Communication is an effective mechanism for coordinating the behavior of multiple agents. In the field of multi-agent reinforcement learning, agents can improve the overall learning performance and achieve their objectives by communication. Moreover, agents can communicate various types of messages, either to all agents or to specific agent groups, and through specific channels. With the growing body of research work in MARL with communication (Comm-MARL), there is lack of a systematic and structural approach to distinguish and classify existing Comm-MARL systems. In this paper, we survey recent works in the Comm-MARL field and consider various aspects of communication that can play a role in the design and development of multi-agent reinforcement learning systems. With these aspects in mind, we propose several dimensions along which Comm-MARL systems can be analyzed, developed, and compared.

KEYWORDS

Multi-Agent Learning, Reinforcement Learning, Communication, Survey

ACM Reference Format:

Changxi Zhu, Mehdi Dastani, and Shihan Wang. 2022. A Survey of Multi-Agent Reinforcement Learning with Communication. In *Proceedings of ACM Conference 2022.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Many real world scenarios involve multiple agents interacting and affecting a common environment, such as autonomous driving [1], sensor networks [2], robotics [3], and game playing [4, 5]. These problems can be solved by Multi-Agent Reinforcement Learning (MARL), where agents employ reinforcement learning (RL) techniques to develop either cooperative, competitive or mixed of cooperative and competitive behaviours. As agents are often distributed in the environment, nowadays partial observability becomes an essential assumption in MARL [6–8], in which agents have only access to their local observations rather than the state of the environment. Moreover, MARL suffers more from non-stationary issue [9], since each agent not only faces with a changeable environment, but also can be influenced by the changing and adapting policies of other agents. Nevertheless, agents can communicate their information such as observations, intentions or experience to stabilize learning. With communication, agents will have a better understanding of the environment (or the other agents), and are therefore able to coordinate their behaviours [9, 10].

ACM Conference 2022, 2022
2022. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

In this paper, we focus on how communication can be utilized to improve RL agents' learning in the environment. Specifically, we focus on learnable communication protocols, by relaxing the assumption that they are given and fixed. This is aligned with recent works that emphasize on developing dynamic and adaptive communication including learning when, how and what to communicate with powerful deep reinforcement learning techniques [11–14]. However, as the research field MARL augmented with communication (Comm-MARL) is rapidly expanding, and despite a multitude of surveys in this field [10, 15–19], there is lack of a systematic and structural methodology to distinguish and classify Comm-MARL systems. Such a methodology would guide the design and development of MARL systems. Imagine that we plan to develop a new Comm-MARL system for tasks at hand. From the problems of when, how and what to communicate, the system can be characterized with various aspects: agents need to learn whom to communicate with, when to communicate, which piece of information to communicate, how to combine and integrate received information, and finally what kind of learning goals should be achieved with the aid of communication.

We thereby propose 9 dimensions that correspond to unique aspects of Comm-MARL systems. These aspects form the skeleton of a Comm-MARL system and need to be analyzed and compared thoroughly, in order to discover the insights of design. In this way, we aim to identify various aspects of communication that can play a role in the design and development of MARL systems. By mapping the recent Comm-MARL systems 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-MARL systems.

In Section 2 we briefly summarize the recent developments of Comm-MARL systems, showing the importance of a structural way to distinguish and classify Comm-MARL systems. In Section 3, we present our proposed dimensions, explaining how we group the recent works in the categories of each dimension. In Section 4, driven by the proposed dimensions, we discuss our view on the promising open questions in the area and conclude the survey.

2 RELATED WORK

Seminal works such as CommNet [12], DIAL [11] and RIAL [11] allow learning to communicate between deep reinforcement learning agents in cooperative games with partial observations. CommNet learns a shared neural network for agents to process local observations. Each agent's decisions depend on observations and a mean vector of messages (i.e., hidden layers) from other agents, which potentially views other agents in equal. In principle, agents can perform in the environment in a decentralized way with the copies of the shared neural network, while instantaneous communication with all agents is required. In RIAL and DIAL, each agent learns to share a binary or a real value message, which is suitable for limited communication. DIAL integrates the learning of communication and environment policy as a unit and enables gradients to flow across agents. This training paradigm is so called *end-to-end* and has been followed by many works [13, 14, 20–22]. In case that communicated messages are discrete values and thus gradients are not able to be calculated, RIAL utilizes another RL algorithm to learn the content of messages, as we can see in recent works [23–27]. CommNet, DIAL and RIAL are evaluated on fully-cooperative environments with a low number of agents. They utilize a fully connected structure among agents and leave the problem of how to communicate more efficiently and effectively as an open question. Recent works have targeted the problems of when, what and with whom to communicate, based on a predefined or learned communication structure, with possible realistic constraints.

Earlier works utilize a gate mechanism for each agent to decide whether to communicate their messages. ATOC [23] is proposed to communicate with certain agents in an observable filed. Only nearby agents are able to participate in a communication group, which is determined by a probabilistic gate mechanism. Within a communication group, a bi-LSTM is used to automatically combine messages sent from each agent and to send back to each member. IC3Net [24], extended from CommNet, also uses a gate mechanism, while deterministically decides to send a message to either all agents or to no agent at all. In addition, IC3Net employs individualized rewards for each agent rather than a globally shared reward as used in CommNet, thus showing more diverse behaviours in competitive/mixed environment. ETCNet [27] also uses a gate for each agent to decide whether to broadcast their messages. However, the overall probability of sending messages is regularized by a penalty term during optimization, in order to reduce communication overhead. I2C [28] measures casual effect of considering other agents' actions on each agent's own strategy. Then each agent decides whether to communicate with others in a peer-to-peer way.

Although a gate unit provides flexible decisions for communication, we could prioritize the chances of communication or explicitly establish a communication graph in a global way. SchedNet [25] learns to choose a certain number of agents to broadcast their messages. GA-Comm [22], MAGIC [13] and FlowComm [14] learn a shared graph for agents to decide whether and with whom to communicate. GA-Comm learns a undirected communication graph by using attention mechanism to decide which pair of agents can communicate with each other. In contrast, MAGIC and FlowComm generate more fine-grained control by building directed graph among agents. Then connected agents can communicate with others unilaterally or bilaterally.

Some works use predefined relations among agents to decide when and with whom to communicate, while learning the content of messages. Agent-Entity Graph [29] utilizes a pretrained graph to maintain relations between agents. Then connected agents will communicate their individual encoding of observations and observable entities in the environment. Network Communication [30, 31] builds upon networked multi-agent systems (NMASs), where decentralized agents are sparsely connected by a predefined communication network. Here we consider explicit messages transferring among agents during both training and execution in a NMAS. NeurComm [32] assumes a spatiotemporal MDP, where transitions depend only on neighboring agents. Then decentralized agents communicate local states, action probabilities, and their own belief states with neighbors to improve observability and reduce non-stationarity. Finally agents will act based on updated belief states. IP [33] assumes a similar decomposition of the task, where each agent's reward will only be affected by its own action and the actions of neighbors. Then agents broadcast and coordinate their policy upon the communication network.

Most works leverage local information to generate an encoding as the content of messages. The encoding may contain individual observations [11–14, 20–22, 25], or intended actions (or plans) [23, 34]. Received messages can be concatenated together to prevent information loss [11, 21, 25, 27]. Agents could also explicitly send signatures with messages to inform other agents how to address the importance of messages. TarMAC [20] and IMMAC [35] use a broadcast way for each agent to send messages with signatures. Agents who receive messages assign weights to each message by considering associated signature. Different from IMMAC, TarMAC employs an attention mechanism to produce weights while IMMAC uses softmax. The attention mechanism is also exploited in GA-Comm [22] due to the flexibility and strength. GA-Comm learns two attention layers, one for deciding whether to communicate with others, and another for determining the relative importance between agents, together with a GNN network to aggregate messages. Nevertheless, the importance of messages can be learned via neural network implicitly. BiCNet [36] proposes to connect each agent's policy and value function by bi-LSTM layers. Thus agents are able to capture others' memory states with long term dependency and exchange messages accordingly. MD-MADDPG [37] allows agents to maintain a shared memory, which serves as the context of their world. Then agents learn to sequentially read and write the memory as in LSTM. DGN [38] utilizes convolutional layers to attain (latent) features from neighboring agents.

A robust Comm-MARL system needs to adapt for realistic constraints such as costly communication, stochastic environments, etc. SchedNet [25] considers the issues of communication overhead and contention under the constraints of limited bandwidth and a shared communication channel. Thus only a limited number of agents are selected to send messages to the channel. VBC [39] proposes to limit the variance of the transferred messages by a threshold to filter higher-variance messages so that achieves lower communication overhead. TMC [40] disallows agents who produce similar messages within a time window to broadcast their messages. Then those messages which are fairly different from messages sent before will be shared to all agents. Moreover, TMC utilizes a message buffer to store received messages to compensate missing messages. Gated-ACML [26] proposes to actively prune messages in two steps. The first step is similar to ATOC and IC3Net, which learns a gate mechanism to choose whether to send messages or not. In the second step, however, Gated-ACML assumes a centralized message coordinator that coordinates messages and sends back to each agent. Communication can be reduced since in theory each agent only needs to communicate with one another, that is the coordinator. Inspired from information theory [41, 42], IMAC [21] and ETCNet [27] formalise limited bandwidth as optimization constraints. IMAC claims that limited bandwidth requires agents to send low-entropy

Table 1: Designing a Comm-MARL system step by step.

Dimensions	*Targeted Problems*
Communication Type	Which type of agents to communicate with?
Communication Policy	When and how to build communication links among agent?
Communicated Messages	Which piece of information to share?
Message Combination	How to combine received messages?
Inner Integration	How to integrate combined message into learning models?
Communication Constraints	How to fulfill realistic requirements?
Communication Learning	How to train and improve communication?
Training Scheme	How to utilize collected experience from agents?
Controlled Goals	What kind of behaviours are desired to emerge with communication?

Table 2: The category of communicatee type.

Types	*Subtypes*	*Methods*
Agents in the MAS	Nearby Agents	DGN [38]; MAGNet-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]; MADDPG-M [47]; IC3Net [24]; SchedNet [25]; DCC-MD [48]; VBC [39]; Diff Discrete[49]; I2C [28]; IS [34]; ETCNet [27]; Variable-length Coding [43]; TMC [40];
Proxy		MS-MARL-GCM [50]; ATOC [23]; MD-MADDPG [37]; IMAC [21]; GA-Comm [22]; Gated-ACML [26]; HAMMER [51]; MAGIC [13];

messages and propose to clip messages' variance. ETCNet deduces an upper bound of the probability that agents are allowed to send messages at each step, and then optimizes under the constraint of limited bandwidth. Variable-length Coding [43] also considers the case of limited bandwidth, while agents regulate the number of bits they send at a given time step.

3 PROPOSED DIMENSIONS

We aim at illustrating a systematic and structural way of designing a Comm-MARL system. We propose that the system can be characterized by 9 dimensions. The dimensions and their corresponding target problems are depicted in Table 1. In the following subsections, we will summarize recent works and classify them through the proposed dimensions.

3.1 Communicatee Type

Communicatee Type determines which type of agents will potentially receive messages in a Comm-MARL 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 is composed of agents in the environment, which means that agents will directly communicate with each other. Nevertheless, due to partial observablity, agents may not communicate with every agent in the MAS and thus we further distinguish the agents as follows:

- **Nearby Agents**. In many MARL systems, the communication is only allowed between neighbors. Nearby agents can be defined as observable agents [44], agents within a certain distance [29, 38, 45] or neighboring agents on a graph [32]. GAXNet [44] labels agents who are observable and enables communication between them. DGN [38] limits communication within 3 closest neighbors while using a distance metric to find them. Agent-Entity Graph [29] also uses distance to measure nearby agents while as long as two agents are close to each other will be allowed to communicate. LSC [45] enables agents within a cluster radius to decide whether to become a leader agent. Then all non-leader agents in a cluster will communicate with the only one leader agent. NeurComm [32] and IP [33], which are built upon networked multi-agent systems, preset a graph structure among agents. Therefore communicatees will be restricted to neighbors on the graph during learning. MAGNet-SA-GS-MG [46] uses a pretrained graph to limit communication as well. Neighboring agents can also emerge during learning instead of being predefined, as proposed in GA-Comm [22], MAGIC [13] and FlowComm [14], which explicitly learn a graph structure among agents. However, in GA-Comm [22] and MAGIC [13], a central unit (e.g., GNN) learns a graph inside and coordinate messages based on the graph simultaneously. Thus, agents do not communicate directly, and therefore we locate these two works into the category of proxy.
- **Other (Learning) Agents**. In case that nearby agents are not identified, the set of communicatees is simply composed of other (learning) agents. Specifically, IC3Net [24] enables communication between learning agents and their opponents (with fixed policies). Experiments show that opponents will finally learn to not communicate to prevent being exploited.

Proxy. A proxy is a visual agent who plays an essential role in communication, but does not have direct effect on the environment. Using a proxy as a communicatee means that agents will not directly communicate with each other while viewing the proxy as a medium, which could coordinate and transform messages for certain purpose. MS-MARL-GCM [50] utilizes a master agent who collects local observations and hidden states from agents in the environment and sends a common message back to each of them. Similar to MS-MARL-GCM, HAMMER [51] employs a central proxy who gathers local observations from agents in the MAS while sending a private message to each agent. ATOC [23] uses a LSTM to connect nearby agents who decide to participate in a communication group, and a coordinated messages will be shared to each member. MD-MADDPG [37] maintains a shared memory among agents and learns to selectively store local observations into the memory and load the memory. IMAC [21] defines a scheduler, which aggregates encoded information from all agents and sends individual messages to each agent. However, in Gated-ACML [52], agents will decide whether to communicate with a message coordinator. GA-Comm [22] and MAGIC [13] learns a global message processor to integrate messages from agents based on their weights.

Table 2 summarizes recent works on communicatee type. To demonstrate above categories, we present a vivid example of how different communicatee types are used in a Comm-MARL system in Figure 1. As we can see, agent 3 and agent 4 are nearby agents of

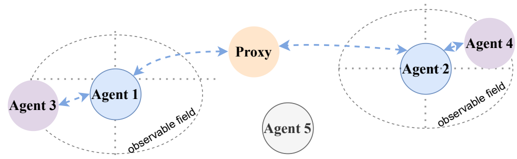


Figure 1: Three communicatee types in the same system.

agent 1 and agent 2 respectively, while agent 5 is another agent who is beyond the view of agents 1 and 2. Then, agent 1 can communicate with its neighboring agent 3 or the proxy if possible.

3.2 Communication Policy

Communication Policy defines how to make the decisions of whether to communicate with potential communicatees to enable message transferring. A communication policy can be either predefined or learned. In a predefined structure, we may allow full communication without considering whether each pair of agents should communicate, or using fixed parameters such as averaging number of communicated agents during learning to enable a more dynamic communication structure. Despite this, learning to determine how to build communication structure among agents offers generalization abilities to more scenarios, and it becomes more and more popular due to its flexibility. A learnable communication policy can be undertaken by local agents, or shared by the entire MAS. Furthermore, if a proxy is presented, agents can individually learn to communicate with the proxy [23, 26]. The central proxy itself can also form a global communication structure to enable both messages coordination and efficient communication [13, 22]. Based on these observations, we categorize current works on determining a communication policy into four categories (summarized in Table 3). We also present examples of how agents build communication links in four types of communidcation policy in Figure 2. Both Full Communication and Partial Structure use a predefined communication policy to decide whether to communicate. In contrast, Individual Control and Global Control learn a local communication policy and a global communication policy respectively, to build communication links between agents (or a possible proxy). If a proxy is presented (usually to be central), the proxy coordinates messages from agents who decide to communicate with the proxy. The categories and associated works are summarized as follows:

Full Communication. In this category, each pair of agents will be connected and messages are transmitted in a broadcast manner. Full communication can be viewed as a fully connected structure, which is often used in early works on Comm-MARL. DIAL [11], RIAL [11], CommNet [12], and BiCNet [36] learn a communication protocol which connect all agents together. Diff Discrete [49] and Variable-length Coding [43] consider two-agent cases while do not learn to block messages from each other. TarMAC [20] and IS [34] learn meaningful messages while using a broadcast way to share messages thus still using full communication. DCC-MD [48] drops out received messages with a certain probability to reduce input dimension while does not learn whether to communicate. In IMAC [21], MS-MARL-GCM [50] and HAMMER [51], a central proxy who receives local observations or encoded messages always connects agents in the MAS.

Table 3: The category of communication policy

Types	*Subtypes*	*Methods*
Predefined	Full Communication	DIAL [11]; RIAL [11]; CommNet [12]; BiCNet [36]; MS-MARL-GCM [50]; TarMAC [20]; MD-MADDPG [37]; DCC-MD [48]; IMAC [21]; Diff Discrete[49]; IS [34]; Variable-length Coding [43]; HAMMER [51];
	Partial Structure	DGN [38]; MAGNet-SA-GS-MG [46]; Agent-Entity Graph [29]; VBC [39]; NeurComm [32]; IP [33]; TMC [40]; GAXNet [44];
Learnable	Individual Control	ATOC [23]; MADDPG-M [47]; IC3Net [24]; Gated-ACML [26]; LSC [45]; I2C [28]; ETCNet [27];
	Global Control	SchedNet [25]; GA-Comm [22]; MAGIC [13]; FlowComm [14];

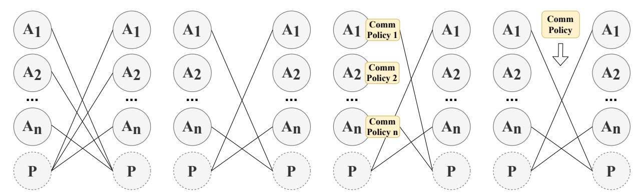


Figure 2: Four types of communication policy with agents (shown as A) in the environment and a possible proxy (shown as P).

(Predefined) Partial Structure. The communication relations between agents can be captured by a predefined graph which dynamically changes over time. Then each agent communicate with a limited number of agents rather than all the others. NeurComm [32] and IP [33] are based on a networked multi-agent system, by randomly generating a communication network while keeping the average number of communicated agents during learning being fixed. DGN [38], MAGNet-SA-GS-MG [46], and GAXNet [44] only allow communication within a certain number of nearby agents. Agent-Entity Graph [29] utilizes a pretrained graph to capture relations between agents. VBC [39] and TMC [40] use a handcrafted threshold to filter the chance of communication.

Individual Control ($\mathcal{I}_c$). Each agent actively and individually decides whether to communicate with other agents. The communication links between agents implicitly form a graph structure. Most works utilize a learnable gate mechanism to facilitate the decision of communication. IC3Net [24] and ATOC [23] introduces a gate mechanism for agents to decide whether to broadcast their messages in a deterministical way and a probabilistic way, respectively. I2C [28] learns to bilaterally decide whether to communicate with each of the other agents, by evaluating the effect of other agents on an agent's own strategy. ETCNet [27] also uses a gate unit while limiting the total probability of sending message behaviours. If a proxy is presented, e.g., a message coordinator, Gated-ACML [26] learns a gate mechanism for each agent to decide whether to communicate with the proxy, instead of directly communicating with other agents. Different from using a gate function, LSC [45] allows each group of agents (in a certain range) to compare their weights to be a leader agent, and then enable communication from each group to their leaders, and from leader to leader [1].

Global Control. A globally shared communication policy can be learned to endow more precise control of communication links between agents. SchedNet [25] learns a global scheduler which

[1]The leader agent is not viewed as a proxy as it still acts in the environment.

Table 4: The category of communicated messages.

Types	*Methods*
Existing Knowledge	DIAL [11]; RIAL [11]; CommNet [12]; BiCNet [36]; MS-MARL-GCM [50]; DGN [38]; TarMAC [20]; MAGNet-SA-GS-MG [46]; MADDPG-M [47]; IC3Net [24]; MD-MADDPG [37]; SchedNet [25]; DCC-MD [48]; Agent-Entity Graph [29]; VBC [39]; IMAC [21]; GA-Comm [22]; Gated-ACML [26]; LSC [45]; Diff Discrete[49]; I2C [28]; ETCNet [27]; Variable-length Coding [43]; TMC [40]; HAMMER [51]; MAGIC [13]; FlowComm [14]; GAXNet [44];
Imagined Future Knowledge	ATOC [23]; NeurComm [32]; IP [33]; IS [34];

Table 5: The category of message combination.

Types	*Methods*
Concatenation	DIAL [11]; RIAL [11]; MADDPG-M [47]; SchedNet [25]; Diff Discrete[49]; IS [34]; ETCNet [27]; Variable-length Coding [43];
Equally Valued	CommNet [12]; IC3Net [24]; VBC [39]; FlowComm [14];
Unequally Valued	BiCNet [36]; MS-MARL-GCM [50]; ATOC [23]; DGN [38]; TarMAC [20]; MAGNet-SA-GS-MG [46]; MD-MADDPG [37]; DCC-MD [48]; Agent-Entity Graph [29]; IMAC [21]; GA-Comm [22]; Gated-ACML [26]; LSC [45]; NeurComm [32]; IP [33]; I2C [28]; TMC [40]; HAMMER [51]; MAGIC [13]; GAXNet [44];

only allows a certain number of agents to broadcast their messages to reduce communication. FlowComm [14] learn a directed graph among agents, and therefore agents can communicate with each other unilaterally or bilaterally. Similarly, GA-Comm [22] and MAGIC [13] learn an undirected graph and a directed graph respectively. However, they introduce extra message coordinator to coordinate and transform messages from agents.

3.3 Communicated Messages

Once building communication links among agents via a communication policy, agents should decide which piece of information will be transferred among them. Due to the popular assumption of partial observability, local observations become vital for coordination. Furthermore, agents can leverage historical experience, or their intended actions or future plans to generate more informative messages. Based on whether future information is simulated and encoded, we classify recent works in this dimension into the following two categories (summarized in Table 4).

Existing Knowledge. In this category, agents utilize existing knowledge (e.g. past observations or actions) to facilitate communication, and most recent works use an encoding of these knowledge as messages. In particular, the family of RNN (e.g., LSTM and GRU) is often employed as the encoding function which is able to selectively forget and store historical observations [12–14, 20, 22, 24, 28, 36, 50], or action-observation histories [11, 36]. Nevertheless, if a proxy is presented, messages will be generated and transformed from agents to the proxy, and then from the proxy to agents. Therefore we differentiate recent works on using existing knowledge as messages into the following two cases.

- **With Proxy**. With using a proxy, communicated messages will be generated by two steps. Firstly, local observations can be encoded [13, 21, 22, 26, 37] or directly sent [50, 51] to the proxy. Then, the proxy who gathers local (encoded) observations can generate a single new message to all agents [50], or individualised messages to each agent [13, 21, 22, 26, 37, 51]. Both ways produce a message containing global information and agents do not need to make any efforts to consider how to combine messages.
- **Without Proxy**. Without using a proxy, messages are directly sent to each agent. DIAL and RIAL [11] use an encoding of past observations and actions, and local observation as messages. BiCNet [36] is not only feed into local view of each agent but also a global observation of the environment. Other works directly communicate observations [47], or use simple feed-forward network [25, 27, 43, 49], MLP [39, 40, 46], autoencoder [48], CNN [38], RNNs [12, 14, 20, 24, 28] or GNN [29, 45] to obtain encoded local observations. In addition, agents could communicate more specific information, for example, in GAXNet [44], agents coordinate their local attention weights used for combining hidden states from neighboring agents.

Imagined Future Knowledge. We refer Imagined Future Knowledge to either intended actions [23], policy fingerprint (i.e., current action probabilities in a particular state) [32, 33], or future plans [34], which can be generated by simulating a model of environment dynamics [34]. As intentions and plans are state-related, recent works usually encode intentions together with local observations to generate more relevant messages.

3.4 Message Combination

Current works on Comm-MARL usually process received (multiple) messages as a whole. Message Combination determines how to combine received messages before feeding them into agents' internal model. If a proxy is presented, generally each agent receives coordinated and combined messages from the proxy, excluding the necessary to operate message combination, as discusses in the dimension of Communicated Messages. If no proxy is presented, each agent will then individually decide how to combine more than one message. Since communicated messages encode the senders' personal understanding about the learning or the environment, some messages may be more valued than others. As shown in Table 5, we therefore categorize recent works in this dimension according to agents' preference on messages.

Concatenation. Messages are concatenated thus no preference is introduced. By concatenation, information will not be lost despite expanding input space [11, 25, 27, 34, 43, 47, 49]. Therefore messages are either represented as single values [11, 27, 43] or short vectors [25], and are experimented on only a few agents.

Equally Valued. When agents who send messages are potentially treated as the same, messages are combined equally. A final message can be generated by averaging [12, 24, 39] or summing up [14] received messages (vectors).

Unequally Valued. In this category, agents as well as their messages are valued differently. Handcrafted rules are used in DCC-MD [48] and TMC [40] to prune some received messages. Nevertheless, Attention mechanism is often used to assign weights to each received message and then combine them together [20, 29, 46]. In addition, the combination of messages can be modelled as a neural network, which implicitly imposes preferences on messages. Simple neural network [21, 26, 51], CNN [38], LSTM (or RNN) [23, 28, 32, 36, 37, 44, 50], and GNN [13, 22, 33, 45] are used to automatically learn to combine messages. Among them, GNN utilizes the learned graph structure of agents and can be combined with attention mechanism to assign weights to neighboring agents.

Table 6: The category of inner integration.

Types	*Methods*
Policy-level	CommNet [12]; MS-MARL-GCM [50]; ATOC [23]; MAGNet-SA-GS-MG [46]; IC3Net [24]; MD-MADDPG [37]; SchedNet [25]; IMAC [21]; GA-Comm [22]; Gated-ACML [26]; Diff Discrete[49]; IP [33]; I2C [28]; IS [34]; ETCNet [27]; Variable-length Coding [43]; HAMMER [51]; FlowComm [14]; GAXNet [44];
Value-level	DIAL [11]; RIAL [11]; DGN [38]; DCC-MD [48]; VBC [39]; LSC [45]; TMC [40];
Policy-level and Value-level	BiCNet [36]; TarMAC [20]; MADDPG-M [47]; Agent-Entity Graph [29]; NeurComm [32]; MAGIC [13];

Table 7: The category of communication constraints.

Types	*Methods*
Limited Bandwidth	DIAL [11]; RIAL [11]; SchedNet [25]; VBC [39]; IMAC [21]; Gated-ACML [26]; ETCNet [27]; Variable-length Coding [43]; TMC [40];
Noisy Channel	DIAL [11]; Diff Discrete[49];
Shared Medium	MD-MADDPG [37]; SchedNet [25];

3.5 Inner Integration

Inner Integration determines how to integrate (combined) messages into an agent's learning model. As most of the literature views messages as additional observations, agents can take messages as an extra input to a policy function, to a value function or to both. Then we classify recent works on inner integration into the following categories based on which a part of the learning model will be used to integrate messages (summarized in Table 6).

Policy-level. Incorporating messages into a policy model is equivalent to say, agents will choose the next action conditioning on received messages. By exploiting information from other agents, each agent will no longer act independently. The policy can be learned by policy gradient with REINFORCE [12, 22, 24, 50], which collects rewards in episodes and trains model after finishing each episode, or actor-critic methods [14, 21, 23, 25–28, 33, 34, 37, 43, 44, 46, 49, 51], which assumes that a critic models (i.e., a Q-function) guides the learning of an actor model (i.e., a policy network).

Value-level. In this category, messages are taken as input to a value function (or an action-value function). Most works fall into this part using DQN-like methods [11, 38–40, 45, 48].

Policy-level & Value-level. Using a policy model and a value model together to integrate messages is usually based on actor-critic methods. The received messages can be considered as extra input [29, 36] to feed into the actor model and the critic model separately. The messages can also be combined with local observation to generate new internal states to be shared with both the actor and the critic models [13, 20, 32, 47].

3.6 Communication Constraints

Realistic concerns such as communication cost and noisy environment impair Comm-MARL systems to embrace applications more than simulations. Communication constraints determine which and how real world limitations are handled in a Comm-MARL system. We categorize recent works in this dimension into three types (summarized in Table 7).

Limited Bandwidth. In this category, communication bandwidth and capacity are assumed to be limited. Early works focus on transmitting succinct message to avoid communication overhead. RIAL and DIAL [11] are proposed to communicate a binary message or a real value between two agents respectively, in order to alleviate limited channel capacity. SchedNet [25] jointly considers a shared channel and limited bandwidth. Thus only a subset of agents will be chosen to broadcast their messages according to their importance. VBC [39] and TMC [40] reduce communication cost by using predefined thresholds to filter unnecessary communication, and both show lower communication overhead compared to SchedNet. Gated-ACML [26] learns a probabilistic gate unit to block messages transmitting between agents and a centralized messages coordinator, with the cost of learning to adjust the gate compared with handcrafted thresholds. IMAC [21] explicitly models bandwidth limitation in optimization, which requires agents to send low-entropy messages instead of blocking communication. Inspired by Gated-ACML and IMAC, ETCNet [27] establishes a constrained model to transform bandwidth into a penalty threshold to restrict sending behaviours. Variable-length Coding [43] also utilizes a penalty term while encouraging short messages.

Noisy Channel. In this category, the messages transmitted between agents can be changed due to environment noise. DIAL considers Gaussian noise and shows that adding noise to real valued messages will change the distribution of messages. Diff Discrete[49] also attends to a noisy channel while aiming to backpropagate derivatives through discretized real value messages. With the proposed technique, sending (real value) messages becomes mathematically equivalent to the discretized signal with additive noise, and thus gradients are able to be derived.

Shared Medium. This category considers a contention issue as messages are transmitted by only one medium. MD-MADDPG [37] allows agents to sequentially access a shared memory space to avoid conflicts. SchedNet [25] chooses agents with high importance to broadcast their messages.

3.7 Communication Learning

Communication Learning concentrates on how to update and adjust a communication protocol, including learning a communication policy and the content of messages. The learning of communication can utilize defined feedback (i.e., rewards), which will be maximized by another (reinforcement) learning procedure in addition to the learning of environment policy, or allow gradients backpropagating from one agent to another to provide richer and dense feedback. Nevertheless, using backpropagation requires messages and communication behaviours to be differential, and may be infeasible if agents learn discrete sending behaviours (i.e., sending or not). We categorize recent works in this dimension based on how communication feedback is exploited (summarized in Table 8).

Reinforced. In this category, another (reinforcement) learning algorithm is employed to train the communication protocol. RIAL [11] and HAMMER [51] focus on learning the content of messages in a full communication structure, which do not consider the problem of whether to communicate. Other works [23–28, 45, 47] jointly take the learning of communicated messages and whether to communicate into consideration. In addition, most works [11, 24, 25,

Table 8: The category of communication learning.

Types	*Methods*
Reinforced	RIAL [11]; ATOC [23]; MADDPG-M [47]; IC3Net [24]; SchedNet [25]; Gated-ACML [26]; LSC [45]; I2C [28]; ETCNet [27]; HAMMER [51];
Differentiable	DIAL [11]; CommNet [12]; BiCNet [36]; MS-MARL-GCM [50]; DGN [38]; TarMAC [20]; MAGNet-SA-GS-MG [46]; MD-MADDPG [37]; DCC-MD [48]; Agent-Entity Graph [29]; VBC [39]; IMAC [21]; GA-Comm [22]; Diff Discrete[49]; NeurComm [32]; IP [33]; IS [34]; Variable-length Coding [43]; TMC [40]; MAGIC [13]; Flow-Comm [14]; GAXNet [44];

27, 45, 47, 51] use environment rewards to learn communication. By contrast, ATOC [23] and Gated-ACML [26] propose to use the difference of Q-values between actions generated with and without communication, to label a valuable message if the difference is higher than a threshold. Then a classification task is performed for deciding whether to communicate. Similar to ATOC and Gated-ACML, I2C [28] also trains a classifier to determine whether to communicate, while using the casual effect between two agents instead of the Q-difference, together with a threshold to tag effective communication.

Differentiable. In this category, communication is improved by backpropagating gradients from communicatees. In case that a communication policy is predefined, such as by full communication [11, 12, 20, 21, 34, 36, 37, 43, 48–50], by communicating with nearby agents [29, 32, 33, 38, 44, 46], or by a fixed threshold to filter communication (do not train a classifier) [39, 40], agents will only learn the content of messages via backpropagation. More recent works [13, 14, 22] circumvent non-differentiable communication behaviours by using biased gradient estimators such as Gumbel-softmax trick [53], which requires additional parameter tuning. Specifically, Diff Discrete [49] reconstructs real value messages though received discretized messages and (independent) random channel noise, which is able to estimate and calculate derivatives.

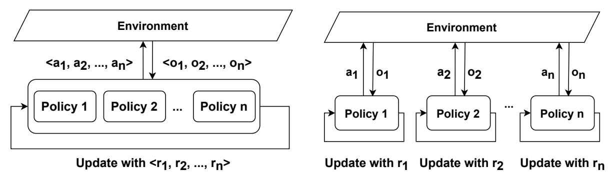
3.8 Training Scheme

This dimension determines how to utilize collected experience (i.e., observations, actions, rewards, and messages) from agents in a Comm-MARL system. We can use a decentralized way to train each agent's model with respective experience. The agents can also be trained centrally with accessing all agents' experience as a whole, and one single model will finally be obtained to control all agents. However, both decentralized and centralized learning suffer from their own issues. Decentralized learning needs to cope with a non-stationary environment due to changing and adapting agents. In contrast, centralized learning faces with a stationary environment while the large joint policy space could be too difficult to search. As a compromising way, centralized training and decentralized execution (CTDE) [11, 54] gradually becomes a standard training scheme for MARL, where agents learn their local policies while using the guidance from central information. We categorize recent works on training scheme according to how agents' experiences are utilized (summarized in Table 9).

Centralized Learning. As shown in Figure 3a, experiences are gathered into a central unit then learning to control all agents. Based on our observations, recent works do not assume a central controller during performing in the environment.

Table 9: The category of training scheme.

Types	*Subtypes*	*Methods*
Decentralized Learning		MAGNet-SA-GS-MG [46]; MADDPG-M [47]; DCC-MD [48]; Agent-Entity Graph [29]; NeurComm [32]; IP [33];
CTED	Individual Parameters	MS-MARL-GCM [50]; SchedNet [25]; IMAC [21]; Gated-ACML [26]; GAXNet [44];
	Parameter Sharing	DIAL [11]; RIAL [11]; CommNet [12]; BiCNet [36]; ATOC [23]; DGN [38]; TarMAC [20]; IC3Net [24]; VBC [39]; GA-Comm [22]; LSC [45]; Diff Discrete[49]; I2C [28]; ETCNet [27]; Variable-length Coding [43]; TMC [40]; HAMMER [51]; MAGIC [13]; FlowComm [14];
	Concurrent	MD-MADDPG [37]; IS [34];



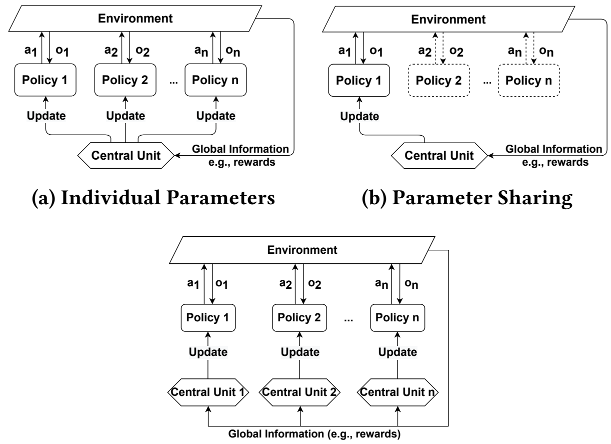
(a) Centralized Learning **(b) Decentralized Learning**

Figure 3: Centralized and Decentralized Learning.

Decentralized Learning. As shown in Figure 3b, experiences are collected individually and agents have independent training processes [29, 32, 33, 46–48].

CTDE. In CTDE, the experiences from all agents are available for optimization. Gradients calculated from all experiences are used to guide the learning of local policies, which enables decentralized execution (in the environment). Moreover, parameter sharing [11] is essential to improve data efficiency, where one set of parameters (i.e., a Q-function or a policy) is shared across agents instead of learning in different processes. Despite this, agents are still able to show various behaviours as they are most likely to receive different observations at each time step. Based on these findings, we further differentiate recent works into the following subgroups.

- **Individual (Policy) Parameters**. In this case, local policies have separate sets of parameters, while a central unit will collect all experiences to provide global information and guidance such as gradients, as shown in Figure 4a. We can use policy gradient algorithm (e.g., with REINFORCE) [50], or actor-critic based methods to train the whole system. [21, 25, 26, 44].
- **Parameter Sharing**. With parameter sharing, all local policies (or local value functions) will use one set of parameters, as shown in Figure 4b. DQN-like algorithms, actor-critic based methods, and policy gradient with REINFORCE are generally used in this situation. If a DQN-like algorithm is employed, a local Q-function will be learned to utilize all experiences [11, 38, 45], or using extra global Q-function to guide the learning [39, 40]. If an actor-critic based method is used, a shared actor (i.e., policy model) will be trained to use all observation-action pairs, and also receives gradient guidance from a central critic [13, 14, 20, 23, 27, 28, 36, 43, 49, 51]. Policy gradient with REINFORCE can be substituted with actor-critic, while requiring sampled rewards in episodes [12, 22, 24].
- **Concurrent**. In case that storing all experiences together is forbidden, agents could make a backup of all experiences if they are assumed to observe other agents' actions and observations, which



(a) Individual Parameters **(b) Parameter Sharing**

(c) Concurrent

Figure 4: Three types of CTDE Scheme.

is different from decentralized learning. Then each local policy will retain individual set of parameters while receive guidance containing global information, as shown in Figure 4c. Concurrent CTDE often uses actor-critic based methods, where each agent has its own central critic to guide its local policy [34, 37].

3.9 Controlled Goal

By specifying a reward configuration, controlled agents are supposed to achieve desired goal and interest. The emergent behaviour of agents can be concluded into three types: cooperative, competitive and mixed [55, 56], which correspond to different reward configurations and learning goals. We note that some works have tested on more than one scenario, to show the flexibility and scalability [13, 20, 24, 38, 45]. We classify recent works in this dimension based on what kind of behaviours among learning agents are desired to emerge with different reward configurations (summarized in Table 10).

Cooperative. In cooperative scenario, agents have the incentive to communicate to achieve better team performance. A team of agents can receive a shared reward [11–14, 20, 21, 23, 25, 26, 28, 29, 34, 39, 40, 44–47, 49–51], which does not account for the contribution of each agent. The agents can also receive local rewards, with extra designs to make the reward depend on teammates' collective performance [22, 24, 36–38, 43, 48, 50, 51], penalize collisions [13, 14, 22, 23, 27, 34, 38, 48], or share reward with neighbors [32, 33] to encourage cooperation.

Competitive. In case that agents need to compete with each other for limit resources or the same object, they are assigned adversary learning objectives, which try to maximize their own cumulative rewards while minimizing the rewards of their opponents. StarCraft [39, 40, 57] is one of the most popular test environments which involves multiple competitive teams. However, most works control one team of agents and are thus beyond our interest. Based on our observations, there is only one work, IC3Net [24], which tests on competitive scenarios with adversary rewards. IC3Net shows that competitive agents learn to communicate only when it is profitable, e.g., before reaching the target.

Table 10: The category of controlled goals.

Types	*Configurations*	*Methods*
Cooperative	Global Rewards	DIAL [11]; RIAL [11]; CommNet [12]; MAGNet-SA-GS-MG [46]; MADDPG-M [47]; SchedNet [25]; Agent-Entity Graph [29]; VBC [39]; IMAC [21]; Gated-ACML [26]; LSC [45]; Diff Discrete[49]; I2C [28]; TMC [40]; GAXNet [44];
	Local Rewards	BiCNet [36]; DGN [38]; IC3Net [24]; MD-MADDPG [37]; DCC-MD [48]; GA-Comm [22]; NeurComm [32]; IP [33]; ETCNet [27]; Variable-length Coding [43];
	Global or Local Rewards	MS-MARL-GCM [50]; ATOC [23]; TarMAC [20]; IS [34]; HAMMER [51]; MAGIC [13]; FlowComm [14];
Competitive	Conflict Rewards	IC3Net [24];
Mixed	Self-interested Rewards	DGN [38]; TarMAC [20]; IC3Net [24]; LSC [45]; MAGIC [13];

Mixed. For a MAS where we care about self-interest agents, individual rewards without depending on other agents can be distributed to each agent [13, 20, 24, 38, 45]. Therefore cooperative and competitive behaviours may coexist during learning. Specifically, DGN [38] considers a game that agent gets positive reward by eating food, but gets higher reward by attacking other agent. With communication, agents can learn collaboratively sharing resources rather than attacking each other. IC3Net [24], TarMAC [20] and MAGIC [13] are evaluated on a mixed version of Predator-prey, and agents learn to communicate only when necessary.

4 DISCUSSION AND CONCLUSION

We identified 9 dimensions to analyze and compare different Comm-MARL systems, from which researchers can develop their own Comm-MARL system. Despite the prosperity of this area, there are still some problems that need to be further considered and solved. First, most recent works on Comm-MARL can be concluded to use a Sender-Receiver or Sender-Proxy-Receiver paradigm to enable communication, that is, by assuming that agents *inform* some knowledge regarding their learning or observations to others. This is convenient for learning since gradients can be easily backpropagated from communicatees. However, agents could *request* specific information from others. For example, Xuan et al. [58] summarize that agents can tell, query or sync their knowledge, which show more types of communication. Second, as reviewed in Section 3.6, communication constraints play an important role for scenarios with the requirements of low communication cost and robust communication, which need to be further explored and combined with realistic applications. Third, evaluating the effect of a communication protocol is intractable, as it is hard to recognize whether performance improvements is due to communicated messages or actions executed in the environment. In the dimension of communication learning, we identify that there are two categories of learning a communication protocol, either Reinforced or Differential. Nevertheless, Reinforced way needs human force to design proper feedback for learning, and Differential way may confront the problem of how each agent contributes to a shared reward. We need to develop more sophisticated and effective way to learn communication. Last but not the least, parameter sharing is very popular in recent works, however, it assumes homogeneous learning models. How to develop a Comm-MARL system for heterogeneous agents is not well explored.

REFERENCES

[1] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *CoRR*, abs/1610.03295, 2016.

[2] Meritxell Vinyals, Juan A. Rodríguez-Aguilar, and Jesús Cerquides. A survey on sensor networks from a multiagent perspective. *Comput. J.*, 54(3):455–470, 2011.

[3] Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *Int. J. Robotics Res.*, 32(11):1238–1274, 2013.

[4] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy P. Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nat.*, 550(7676):354–359, 2017.

[5] Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. *Science*, 365(6456):885–890, 2019.

[6] Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs.* Springer Briefs in Intelligent Systems. Springer, 2016.

[7] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6379–6390, 2017.

[8] Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 2974–2982. AAAI Press, 2018.

[9] Georgios Papoudakis, Filippos Christianos, Arrasy Rahman, and Stefano V. Albrecht. Dealing with non-stationarity in multi-agent deep reinforcement learning. *CoRR*, abs/1906.04737, 2019.

[10] Mohamed Salah Zaïem and Etienne Bennequin. Learning to communicate in multi-agent reinforcement learning : A review. *CoRR*, abs/1911.05438, 2019.

[11] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 2137–2145, 2016.

[12] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pages 2244–2252, 2016.

[13] Yaru Niu, Rohan R. Paleja, and Matthew C. Gombolay. Multi-agent graph-attention communication and teaming. In *20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 964–973, 2021.

[14] Yali Du, Bo Liu, Vincent Moens, Ziqi Liu, Zhicheng Ren, Jun Wang, Xu Chen, and Haifeng Zhang. Learning correlated communication topology in multi-agent reinforcement learning. In *20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 456–464, 2021.

[15] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E. Taylor. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 33(6):750–797, 2019.

[16] Thanh Thi Nguyen, Ngoc Duy Nguyen, and Saeid Nahavandi. Deep reinforcement learning for multi-agent systems: A review of challenges, solutions and applications. *CoRR*, abs/1812.11794, 2018.

[17] Afshin Oroojlooyjadid and Davood Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *CoRR*, abs/1908.03963, 2019.

[18] Sven Gronauer and Klaus Diepold. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, pages 1–49, 2021.

[19] Annie Wong, Thomas Bäck, Anna V. Kononova, and Aske Plaat. Multiagent deep reinforcement learning: Challenges and directions towards human-like approaches. *CoRR*, abs/2106.15691, 2021.

[20] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 1538–1546, 2019.

[21] Rundong Wang, Xu He, Runsheng Yu, Wei Qiu, Bo An, and Zinovi Rabinovich. Learning efficient multi-agent communication: An information bottleneck approach. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 9908–9918. PMLR, 2020.

[22] Yong Liu, Weixun Wang, Yujing Hu, Jianye Hao, Xingguo Chen, and Yang Gao. Multi-agent game abstraction via graph attention neural network. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, pages 7211–7218, 2020.

[23] Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation. In *Advances in Neural Information Processing Systems 31 (NIPS)*, pages 7265–7275, 2018.

[24] Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Individualized controlled continuous communication model for multiagent cooperative and competitive tasks. In *International Conference on Learning Representations (ICLR)*, 2019.

[25] Daewoo Kim, Sangwoo Moon, David Hostallero, Wan Ju Kang, Taeyoung Lee, Kyunghwan Son, and Yung Yi. Learning to schedule communication in multi-agent reinforcement learning. In *7th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2019.

[26] Hangyu Mao, Zhengchao Zhang, Zhen Xiao, Zhibo Gong, and Yan Ni. Learning agent communication under limited bandwidth by message pruning. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 5142–5149. AAAI Press, 2020.

[27] Guangzheng Hu, Yuanheng Zhu, Dongbin Zhao, Mengchen Zhao, and Jianye Hao. Event-triggered multi-agent reinforcement learning with communication under limited-bandwidth constraint. *CoRR*, abs/2010.04978, 2020.

[28] Ziluo Ding, Tiejun Huang, and Zongqing Lu. Learning individually inferred communication for multi-agent cooperation. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

[29] Akshat Agarwal, Sumit Kumar, Katia P. Sycara, and Michael Lewis. Learning transferable cooperative behavior in multi-agent teams. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1741–1743. International Foundation for Autonomous Agents and Multiagent Systems, 2020.

[30] Kaiqing Zhang, Zhuoran Yang, and Tamer Basar. Decentralized multi-agent reinforcement learning with networked agents: recent advances. *Frontiers Inf. Technol. Electron. Eng.*, 22(6):802–814, 2021.

[31] Kaiqing Zhang, Zhuoran Yang, and Tamer Basar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *CoRR*, abs/1911.10635, 2019.

[32] Tianshu Chu, Sandeep Chinchali, and Sachin Katti. Multi-agent reinforcement learning for networked system control. In *8th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2020.

[33] Chao Qu, Hui Li, Chang Liu, Junwu Xiong, James Zhang, Wei Chu, Yuan Qi, and Le Song. Intention propagation for multi-agent reinforcement learning. *CoRR*, abs/2004.08883, 2020.

[34] Woojun Kim, Jongeui Park, and Youngchul Sung. Communication in multi-agent reinforcement learning: Intention sharing. In *9th International Conference on Learning Representations (ICLR)*, 2021.

[35] Chuxiong Sun, Bo Wu, Rui Wang, Xiaohui Hu, Xiaoya Yang, and Cong Cong. Intrinsic motivated multi-agent communication. In Frank Dignum, Alessio Lomuscio, Ulle Endriss, and Ann Nowé, editors, *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, pages 1668–1670. ACM, 2021.

[36] Peng Peng, Quan Yuan, Ying Wen, Yaodong Yang, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games. *CoRR*, abs/1703.10069, 2017.

[37] Emanuele Pesce and Giovanni Montana. Improving coordination in small-scale multi-agent deep reinforcement learning through memory-driven communication. *Machine Learning*, 109(9-10):1727–1747, 2020.

[38] Jiechuan Jiang, Chen Dun, Tiejun Huang, and Zongqing Lu. Graph convolutional reinforcement learning. In *8th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2020.

[39] Sai Qian Zhang, Qi Zhang, and Jieyu Lin. Efficient communication in multi-agent reinforcement learning via variance based control. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 3230–3239, 2019.

[40] Sai Qian Zhang, Qi Zhang, and Jieyu Lin. Succinct and robust multi-agent communication with temporal message control. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33 (NIPS)*, 2020.

[41] Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948.

[42] Roger L Freeman. *Telecommunication system engineering*, volume 82. John Wiley & Sons, 2004.

[43] Benjamin Freed, Rohan James, Guillaume Sartoretti, and Howie Choset. Sparse discrete communication learning for multi-agent cooperation through backpropagation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7993–7998. IEEE, 2020.

[44] Won Joon Yun, Byungju Lim, Soyi Jung, Young-Chai Ko, Jihong Park, Joongheon Kim, and Mehdi Bennis. Attention-based reinforcement learning for real-time UAV semantic communication. *CoRR*, abs/2105.10716, 2021.

[45] Junjie Sheng, Xiangfeng Wang, Bo Jin, Junchi Yan, Wenhao Li, Tsung-Hui Chang, Jun Wang, and Hongyuan Zha. Learning structured communication for multi-agent reinforcement learning. *CoRR*, abs/2002.04235, 2020.

[46] Aleksandra Malysheva, Tegg Tae Kyong Sung, Chae-Bong Sohn, Daniel Kudenko, and Aleksei Shpilman. Deep multi-agent reinforcement learning with relevance graphs. *CoRR*, abs/1811.12557, 2018.

[47] Ozsel Kilinc and Giovanni Montana. Multi-agent deep reinforcement learning with extremely noisy observations. *CoRR*, abs/1812.00922, 2018.

[48] Woojun Kim, Myungsik Cho, and Youngchul Sung. Message-dropout: An efficient training method for multi-agent deep reinforcement learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 6079–6086. AAAI Press, 2019.

[49] Benjamin Freed, Guillaume Sartoretti, Jiaheng Hu, and Howie Choset. Communication learning via backpropagation in discrete channels with unknown noise. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 7160–7168. AAAI Press, 2020.

[50] Xiangyu Kong, Bo Xin, Fangchen Liu, and Yizhou Wang. Revisiting the master-slave architecture in multi-agent deep reinforcement learning. *CoRR*, abs/1712.07305, 2017.

[51] Nikunj Gupta, G. Srinivasaraghavan, Swarup Kumar Mohalik, and Matthew E. Taylor. HAMMER: multi-level coordination of reinforcement learning agents via learned messaging. *CoRR*, abs/2102.00824, 2021.

[52] Hangyu Mao, Zhengchao Zhang, Zhen Xiao, Zhibo Gong, and Yan Ni. 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*, pages 5142–5149. AAAI Press, 2020.

[53] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2017.

[54] Landon Kraemer and Bikramjit Banerjee. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190:82–94, 2016.

[55] Lucian Busoniu, Robert Babuska, and Bart De Schutter. Multi-agent reinforcement learning: A survey. In *Ninth International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1–6. IEEE, 2006.

[56] Laëtitia Matignon, Guillaume J. Laurent, and Nadine Le Fort-Piat. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *Knowledge Engineering Revie*, 27(1):1–31, 2012.

[57] Bo Liu, Qiang Liu, Peter Stone, Animesh Garg, Yuke Zhu, and Anima Anandkumar. Coach-player multi-agent reinforcement learning for dynamic team composition. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 6860–6870. PMLR, 2021.

[58] Ping Xuan, Victor R. Lesser, and Shlomo Zilberstein. Communication decisions in multi-agent cooperation: model and experiments. In Elisabeth André, Sandip Sen, Claude Frasson, and Jörg P. Müller, editors, *Proceedings of the Fifth International Conference on Autonomous Agents, AGENTS 2001, Montreal, Canada, May 28 - June 1, 2001*, pages 616–623. ACM, 2001.