Efficient Communication in Multi-Agent Reinforcement Learning with Implicit Consensus Generation

Dapeng Li[1,2], **Na Lou**[1,2], **Zhiwei Xu**[3], **Bin Zhang***[1,2], **Guoliang Fan***[1,2]

[1]The Key Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]School of Artificial Intelligence, Shandong University
{lidapeng2020, na.lou, zhangbin2020, guoliang.fan}@ia.ac.cn,zhiwei_xu@sdu.edu.cn

Abstract

A key challenge in multi-agent collaborative tasks is reducing uncertainty about teammates to enhance cooperative performance. Explicit communication methods can reduce uncertainty about teammates, but the associated high communication costs limit their practicality. Alternatively, implicit consensus learning can promote cooperation without incurring communication costs. However, its performance declines significantly when local observations are severely limited. This paper introduces a novel multi-agent learning framework that combines the strengths of these methods. In our framework, agents generate a consensus about the group based on their local observations and then use both the consensus and local observations to produce messages. Since the consensus provides a certain level of global guidance, communication can be disabled when not essential, thereby reducing overhead. Meanwhile, communication can provide supplementary information to the consensus when necessary. Experimental results demonstrate that our algorithm significantly reduces inter-agent communication overhead while ensuring efficient collaboration.

Introduction

Multi-agent reinforcement learning (MARL) has garnered increasing research attention in recent years, with applications spanning a wide range of scenarios such as video games (Berner et al. 2019; Vinyals et al. 2019), energy networks (Wang et al. 2021), and robot control (Li et al. 2024a). Many MARL algorithms adopt the centralized training and decentralized execution (CTDE) paradigm due to its scalability and effectiveness in handling non-stationarity in multi-agent environments (Lowe et al. 2017; Foerster et al. 2018; Rashid et al. 2018; Sunehag et al. 2018). Despite the advancements achieved with CTDE, agents can only access local observations during execution, leading to significant uncertainty about the states and information about their teammates. This uncertainty can result in severe discoordination and suboptimal strategies.

To address the above issue, some methods enable explicit communication between agents, such as exchanging agents' local observations or embeddings (Das et al. 2019; Wang et al. 2020; Sukhbaatar, Szlam, and Fergus 2016), and using the received communication messages to enhance local observations for policy selection and learning. These methods can effectively reduce uncertainty among agents and improve cooperative performance. However, high communication costs limit the practicality and effectiveness of these algorithms in real-world applications with strict bandwidth and real-time requirements (Roth, Simmons, and Veloso 2006). Additionally, excessive communication can introduce unnecessary or even harmful information (Zhang, Zhang, and Lin 2019; Guan et al. 2022; Jiang et al. 2020), potentially impairing the convergence of the learning process. Therefore, how to reduce communication costs and send efficient messages becomes a valuable problem.

In contrast to explicit communication, implicit consensus cooperation (Xu et al. 2023; Ruan et al. 2023; Pagello et al. 1999) involves agents inferring a consistent consensus through their respective local observations. By leveraging the consensus, agents can develop a shared understanding of the environment and the group, thereby enhancing cooperative performance without additional communication costs. However, the drawback of implicit consensus is that, due to the limitations of local observations, forming a complex consensus can be challenging when there are significant discrepancies in observations among agents. Furthermore, excessive reliance on consensus may result in a lack of diversity in agent strategies.

Both methods mentioned above have their respective advantages, yet most existing works only consider these two approaches separately, neglecting the potential benefits of their combination. This paper proposes a **CO**nsensus-based **COM**munication algorithm framework called COCOM. In COCOM, each agent infers a consensus in a decentralized manner and subsequently uses its consensus and local observations to generate messages. The entire process can be divided into two main parts: consensus generation and message generation. In the consensus generation part, we design a consensus learning mechanism tailored to the CTDE paradigm. During execution, the consensus module generates local consensus based on individual observation. During training, agents utilize global information to produce a global consensus. This global consensus is used to guide the local consensus, ensuring that the consensus among agents is meaningful and consistent. In the message generation part,

*corresponding author

the message module produces messages based on local consensus and observation, which are then broadcast to the other teammates. Given that all teammates share the same consensus, messages should convey additional information beyond the consensus. We use mutual information to encourage agents to send messages beyond the consensus, and employ a gating mechanism to determine whether the messages need to be sent. The two modules can mutually promote each other: the message module supplements the missing information in consensus, while the consensus module guides the group when communication is not feasible.

We evaluated the performance of COCOM in various multi-agent scenarios, including Hallway Task (Wang et al. 2020), StarCraft Multi-Agent Challenge (Samvelyan et al. 2019) and Google Research Football (Kurach et al. 2020). The experimental results demonstrate that COCOM achieves superior performance with lower communication costs. Additionally, the migration of COCOM into different existing frameworks demonstrates its generalizable improvement effects across various original algorithms. The main contributions of this paper can be summarized as follows:

- We propose a multi-agent collaboration algorithm that combines explicit communication with implicit coordination. Our approach enables agents to infer meaningful consensus and generate differentiated messages, thus enhancing the collective cooperation.
- We design an efficient communication mechanism that filters communication through a gating mechanism based on the messages and consensus. Additionally, we developed a message processing module that uses cross-attention to handle a variable number of teammate messages.
- We conduct extensive experiments on cooperative multi-agent benchmark environments. The results demonstrate that COCOM significantly improves group cooperation performance while reducing communication overhead.

Explicit Communication Coordination

To mitigate the impact of partial observability on agent cooperation, explicit communication methods have been introduced in multi-agent systems. Mainstream approaches focus on learning end-to-end differentiable communication channels. CommNet (Sukhbaatar, Szlam, and Fergus 2016) averages the hidden layer outputs of all agents as a supplement to local observations, but simple averaging ignores inter-agent differences. DAIL (Foerster et al. 2016) broadcasts messages to all teammates and optimizes discrete messages end-to-end via reinforcement learning, but the introduction of discrete messages limits expressiveness. To extract relevant information from a large number of redundant messages, MASIA (Guan et al. 2022) and TarMAC (Das et al. 2019) use attention mechanisms to process teammate messages or observation embeddings. CAMA (Shao et al. 2023) uses a global coach to form a complementary mechanism for agent communication messages. To reduce communication costs, VBC (Zhang, Zhang, and Lin 2019) decides whether to communicate based on confidence in local decisions, but requiring requests and responses from teammates reduces algorithmic real-time performance. TMC (Zhang, Zhang, and Lin 2020) uses temporal locality to reduce communication overhead and improve robustness under noisy conditions, but introducing a communication buffer adds extra storage overhead. CACOM (Li and Zhang 2024) considers generating personalized messages for teammates through two rounds of communication under limited bandwidth. NDQ (Wang et al. 2020) and MAIC (Yuan et al. 2022) use variational inference and regularization penalties to compress and customize messages. ATOC (Jiang and Lu 2018), I2CNet (Ding, Huang, and Lu 2020), and Gated-ACML (Mao et al. 2020b) learn gating mechanisms to control agent communication patterns, but these works lack global coordination of communication messages and overlook information already known to teammates. The common limitation of the above communication methods is that the high communication costs make it difficult for the algorithms to be deployed in practical applications. Although some methods attempt to reduce communication costs by eliminating unnecessary communication, shutting down communication still brings instability to training.

Implicit Consensus Coordination

According to relevant studies in social science, human groups typically form implicit consensus during cooperation (Horowitz 1962; Parikh and Krasucki 1990), where consensus represents an opinion agreed upon by the group. Similarly, in recent years, the collaborative MARL field has conducted a series of studies on consensus formation among agents. Implicit consensus can be achieved through modeling teammates and the group, such as in COLA (Xu et al. 2023), which utilizes the invariant perspective property to learn agents' consensus through comparative learning. During execution, agents infer discrete shared consensus from their respective local observations. However, the consensus generated by COLA is too simplistic. CoS (Ruan et al. 2023) groups agents and uses vectorized variational autoencoders to extract embedded group consensus. TACO (Li et al. 2024b) uses explicit communication to guide implicit cooperation and gradually transitions to a completely communication-free implicit cooperation paradigm. MACKRL (de Witt et al. 2019) constructs common knowledge via hierarchical policy trees and designs joint strategies for agent pairs to utilize this common knowledge. NCC-MARL (Mao et al. 2020a) introduces neighborhood consistency to ensure cognitive alignment among adjacent agents. Although these implicit communication methods significantly improve the practicality of existing communication algorithms, reliance solely on agents' local perceptions of teammates for cooperation in complex dynamic scenarios is often unreliable. Moreover, relying solely on local observations often makes it difficult for agents to reach a more complex consensus among themselves.

In order to alleviate the limitations of the above algorithms, our COCOM algorithm naturally combines the advantages of explicit communication and implicit consensus learning, ensuring that the algorithm can be both effective and practical.

Background

Decentralized Partially Observable Markov Decision Process

This paper considers a fully cooperative multi-agent problem, which can be considered as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) game (Bernstein et al. 2002). A Dec-POMDP can be defined by a tuple $G = \langle S, U, N, \Omega, \mathcal{T}, O, r, \gamma \rangle$, where $s \in S$ is the global state of the environment. At each discrete time step t, agent $i \in N := \{1, \ldots, n\}$ receives an individual partial observation $o_i \in \Omega$ by the observation function $O(s,i) : S \times N \to \Omega$, and select an action $u_i \in U_i$. The joint action $\boldsymbol{u} = \{u_1, \ldots, u_n\} \in \boldsymbol{U} \equiv U^n$ leads to next state s' according to the state transition function $\mathcal{T}(s'|s, \boldsymbol{u}) : S \times U \times S \to P(S)$. $r_i(s, u_i) : S \times U_i \to \mathbb{R}$ is the reward function for each agent i. In Dec-POMDP, all agents share the same reward function $r(s, \boldsymbol{u}) : S \times \boldsymbol{U} \to \mathbb{R}$. The γ is the discount factor. The final goal is to find a joint policy $\boldsymbol{\pi}(\boldsymbol{\tau}, \boldsymbol{u})$ to maximize the global value function $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) = \mathbb{E}_{\boldsymbol{s},\boldsymbol{u}}[\sum_{t=0}^{\infty} \gamma^t r(s, \boldsymbol{u}) | s, \boldsymbol{u}, \boldsymbol{\pi}]$. Here, $\boldsymbol{\tau} = \langle \tau_1, \ldots, \tau_n \rangle$ is joint history representation, and τ_i is the history representation $(o_i^1, u_i^1, \ldots, o_i^{t-1}, u_i^{t-1}, o_i^t)$ of agent i at current timestep t.

Centralized Training and Decentralized Execution

In the CTDE paradigm, agents can utilize all available information during training but only make decisions based on local observations. The value decomposition (VD) method is a representative approach in CTDE, which combines individual utility functions Q_i of all agents to fit the global action value Q_{tot} with a mixing network. Most VD methods follow the assumption of Individual-Global-Max (IGM) which asserts that the overall optimality in multi-agent systems aligns with individual optimality:

$$
\begin{aligned}
&\arg\max_{\boldsymbol{u}} Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u}) \\
&= \left(\arg\max_{u_1} Q_1(\tau_1, u_1), \ldots, \arg\max_{u_n} Q_N(\tau_n, u_n) \right),
\end{aligned}
\tag{1}
$$

The action-value function is trained by minimizing the expected TD error:

$$
\mathcal{L}_{\text{TD}} = (y_{tot} - Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u})), \tag{2}
$$

where $y_{tot} = r + \gamma \max_{\boldsymbol{u}'} \hat{Q}_{tot}(\boldsymbol{\tau}', \boldsymbol{u}')$. $\hat{Q}_{tot}(\cdot)$ is the target network of the joint action-value function.

Method

In this section, we present a detailed description of the COCOM algorithm. COCOM combines the strengths of both explicit communication and implicit consensus. Following QMIX (Rashid et al. 2018), we employ multi-layer perceptron (MLP) and GRU cells (Cho et al. 2014) to obtain the representation of historical information τ_i^t based on local observation o_i^t and the previous action u_i^{t-1}. The core components of COCOM include local and global consensus generators for forming consensus, a message generator for producing messages, and a cross-attention module responsible for receiving messages. The overall framework is shown in Figure 1. Notably, COCOM follows the mainstream design of CTDE and is a general algorithm, making it can be easily integrated into other existing CTDE algorithms. We now provide a detailed description of each module.

Implicit Consensus Learning

The core objective of the consensus problem is to design a protocol or algorithm that enables a group of agents to form a consistent consensus value. Suppose c_i represent the consensus of agent i. Achieving consensus within a team means ensuring that $||c_i - c_j|| \to 0$ as $t \to \infty$ for all $j \neq i$.

To enable decentralized deployment, consensus is generated based on local observations. Existing methods often align consensus values by comparing and minimizing differences between agents' consensus in pairs (Xu et al. 2023). However, this approach can lead to consensus collapsing to trivial values, such as all agents consistently generating the same consensus despite differing local observations. To prevent this, COCOM introduces an additional global consensus generator. The global consensus generator adopts an encoder-decoder structure, as shown in Figure 1(a). Through this structure, we can generate a global consensus and reconstruct the global information according to the consensus, thereby ensuring its significance. The encoder of the global consensus generator takes the global state and joint actions as input and outputs the distribution of the global consensus $q(c^t|s^t, \boldsymbol{u}^t)$. Specifically, the global consensus is represented by a multivariate Gaussian distribution and is calculated by MLP. The global consensus c^t is obtained by sampling from this distribution. The sampled global consensus is then fed into a decoder, also composed of an MLP, to reconstruct the joint action and global state. The learning process of the global consensus generator follows the standard VAE, which minimizes two terms as follows:

$$
\mathcal{L}_{\text{KL}}^{prior} = D_{\text{KL}}(q(c^t|s^t, \boldsymbol{u}^t) || \tilde{p}(\cdot)), \tag{3}
$$

$$
\mathcal{L}_{rec} = \text{CROSS\_ENTROPY}(\boldsymbol{u}^t, \hat{\boldsymbol{u}}^t) + \text{MSE}(s^t, \hat{s}^t), \tag{4}
$$

where D_{KL} is the Kullback-Leibler (KL) divergence, $\tilde{p}(\cdot)$ is the standard Gaussian distribution $\mathcal{N}(0, I)$, $\text{MSE}(\cdot)$ is the mean square error function and $\text{CROSS\_ENTROPY}(\cdot)$ is the cross-entropy loss function. The first term $\mathcal{L}_{\text{KL}}^{prior}$ minimizes the divergence between $q(c^t|s^t, \boldsymbol{u}^t)$ and a uniform prior $\tilde{p}(\cdot)$, and the second term $\mathcal{L}_{rec}$ minimizes the discrepancy between the decoder's output and the encoder's actual input, representing the reconstruction error. The parameters are optimized by reparameterization trick (Kingma and Welling 2014). By reconstructing the global consensus, the generated consensus can provide more valuable information for the group's overall strategy.

The structure of the local consensus generator is similar to the encoder of the global consensus generator, as shown in Figure 1(b). The key difference is that the local consensus generator takes the local history τ_i^t as input and outputs the agent's local consensus distribution $q(\hat{c}_i^t|\tau_i^t)$ through the MLP. Unlike previous methods that compare local consensus between agents in pairs, our approach aligns the local

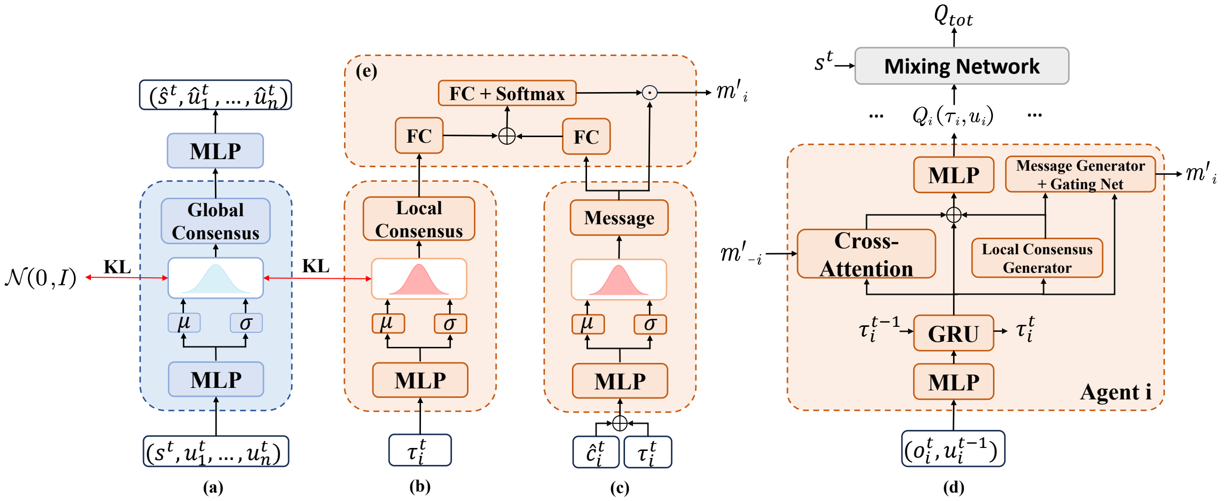


Figure 1: The overall framework of COCOM. (a) Global consensus generator. (b) Local consensus generator. (c) Message generator. (d) Overall architecture. (e) Gating net.

consensus with the global consensus to enhance its meaningfulness. We employ KL divergence to align each agent's local consensus distribution with the global consensus distribution at the same timestep:

$$\mathcal{L}_{\text{KL}} = \sum_{i=1}^{n} D_{\text{KL}}(q(c^t | s^t, \boldsymbol{u}^t) || q(\hat{c}^t | \tau_i)). \tag{5}$$

The global consensus generator is only used during training, while COCOM generates local consensus solely based on local information during execution. Consequently, COCOM can be deployed in a decentralized manner. The overall loss for optimizing the consensus formulation can be summarized as follows:

$$\mathcal{L}_{\text{Cons}} = \mathcal{L}_{rec} + \mathcal{L}_{\text{KL}}^{prior} + \mathcal{L}_{\text{KL}}. \tag{6}$$

Consensus-Based Communication Learning

Since consensus reflects the common understanding among all agents, we expect the message generators to create messages based on the premise of known consensus. To achieve this, we concatenate the local consensus and hidden state as the input of the message generator. The message generator, composed of an MLP, then outputs the message distribution $q(m_i^t | \tau_i^t, \hat{c}_i^t)$. Sampling from this distribution yields the message m_i^t of agent i which is then broadcast to the other agents.

To generate more concise and effective messages, we optimize message generation from two perspectives. Firstly, the messages should avoid duplicating information already in the consensus. Secondly, the message content should aim to reduce the uncertainty in teammates' action selection. Specifically, we optimize message generation by minimizing the following mutual information objective:

$$\mathcal{I}_{\text{Msg}} = \mathcal{I}(m_i; \hat{c}_i | \tau_i) - \mathcal{I}(m_i; u_j | \tau_i, \hat{c}_i), \tag{7}$$

where $\mathcal{I}$ denotes mutual information. Minimizing $\mathcal{I}(m_i; \hat{c}_i | \tau_i)$ ensures that the message is as uncorrelated with the consensus as possible given the agent's history information, thus reducing redundant information already known to teammates. Maximize $\mathcal{I}(m_i; u_j | \tau_i, \hat{c}_i)$ ensures that the message m_i can reduce the uncertainty in teammates' decisions given the history information and consensus, thereby improving the message's effectiveness. By combining these two mutual information terms, the agent's messages will be both concise and effective.

We first address the minimization of the first mutual information term $\mathcal{I}(m_i; \hat{c}_i | \tau_i)$. While directly minimizing conditional mutual information is challenging, existing methods can assist in minimizing its upper bound. We use Conditional-CLUB (Shao et al. 2023) to minimize the upper bound of the first term:

$$\begin{aligned} &\mathcal{I}(m_i; \hat{c}_i | \tau_i) \leq \mathcal{L}_{\mathcal{I}_1} \\ &= \frac{1}{K} \sum_{k=1}^{K} \log p(m_i^k | \tau_i, \hat{c}_i^k) - \frac{1}{K} \sum_{k=1}^{K} \log p(m_i^k | \tau_i, \hat{c}_i^{k'}), \end{aligned} \tag{8}$$

where K is the sample number of a mini-batch, and k' uniformly selected from indices $\{1, 2, \ldots, K\}$.

Next, we discuss the maximization of the second mutual information term $\mathcal{I}(m_i; u_j | \tau_i, \hat{c}_i)$:

$$\mathcal{I}(m_i, u_j | \tau_i, \hat{c}_i) = H(m_i | \tau_i, \hat{c}_i) - H(m_i | \tau_i, \hat{c}_i, u_j). \tag{9}$$

Maximizing this mutual information is equivalent to minimizing the uncertainty in the teammates' decisions after receiving the message, given the agent's local information and consensus. Inspired by the previous work that also maximizes mutual information (Yuan et al. 2022; Wang et al. 2020), we use $q_\zeta(m_i | \tau_i, u_j, \hat{c}_i)$ to approximate the conditional distribution $p(m_i | \tau_i, \hat{c}_i)$ and variationally maximize the lower bound of mutual information as follows:

$$\begin{aligned} &\mathcal{I}(m_i, u_j | \tau_i, \hat{c}_i) \geq -\mathcal{L}_{\mathcal{I}_2} \\ &= -D_{\text{KL}}(p(m_i | \tau_i, \hat{c}_i) || q_\zeta(m_i | \tau_i, u_j, \hat{c}_i)). \end{aligned} \tag{10}$$

Overall, we use the following loss to optimize the mutual information in message generation:

$$\mathcal{L}_{\text{Msg}} = \mathcal{L}_{\mathcal{I}_1} + \mathcal{L}_{\mathcal{I}_2}. \tag{11}$$

Message Pruning with Gating Mechanism

Considering agents can sometimes rely solely on consensus and local observations to make reasonable decisions, we introduce a gating mechanism (Mao et al. 2020b; Li and Zhang 2024) to reduce communication costs by pruning unnecessary messages. The gate mechanism takes the consensus and the generated messages of agent i as inputs:

$$p_g = \text{softmax}(f_g\left(m_i, c_i\right)) = (p_{\text{close}}, p_{\text{open}}), \tag{12}$$

where f_g is the gate network composed of a fully connected (FC) layer, and the output p_g is the probability score of the gate mechanism. p_{open} and p_{close} represent the probabilities of opening and closing the gate. We determine the gating state by comparing p_{open} and p_{close}:

$$g_i = \mathbb{1}[p_{\text{open}} > p_{\text{close}}], \tag{13}$$

where $\mathbb{1}$ is the indicator function. When the p_{open} is greater than p_{close}, which means indicator g_i equals 1, agent i will broadcast its message to all teammates; otherwise, the agent will keep silent. However, the indicator function $\mathbb{1}$ is non-differentiable, gated networks cannot be trained end-to-end using reinforcement learning. To address this issue, we introduce an auxiliary task that enables supervised learning of the gate network by generating pseudo labels. We assess the impact of sending messages on teammates' decision-making to retain valuable communications. Specifically, agent i is permitted to send a message only when it can increase the expected reward of the teammates' chosen action:

$$\begin{aligned} y_i = \mathbb{1}[\frac{1}{n-1}\sum_{j\neq i}(\arg\max_{u_j} Q_j\left(\tau_j, u_j\right)_{|g_i=1} - \\ \arg\max_{u_j} Q_j\left(\tau_j, u_j\right)_{|g_i=0}) > \sigma], \end{aligned} \tag{14}$$

where $Q_j(\tau_j, \cdot)_{|g_i=1}$ denotes the action-value of agent j when agent i choose to sends the message, and $Q_j(\tau_j, \cdot)_{|g_i=0}$ denotes the action-value of agent j when agent i remains silent. We optimize this gating function using cross-entropy loss as follows:

$$\mathcal{L}_g = \text{CROSS\_ENTROPY}(y_i, f_g(m_i, c_i)).$$

Message Receiving Module

Since each agent will receive a varying number of messages at each time step, we introduce a cross-attention mechanism (Vaswani et al. 2017) as the message-receiving module to flexibly handle this variability. Additionally, the attention mechanism enables the agent to focus on more relevant messages. By calculates the attention weight between its historical information τ_i and the received messages, agent i can obtain a fused representation of the received messages:

$$\tilde{m}_i = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad \forall g_j = 1, \tag{15}$$

where $Q = h_i W_Q$, $K = m_j W_K$, $V = m_j W_V$, and W_Q, W_K, W_V is the learnable matrices, and d_k is the dimension of W_K. We use a multi-head attention mechanism to further enhance the diversity of message extraction. When agent i does not receive any message at one timestep, $\tilde{m}_i$ will be filled with zero vectors.

Overall Learning Objective

In order to fully consider known information when making decisions for agents, we concatenate received messages $\tilde{m}_i$, historical information representation τ_i, and the agent's local consensus $\hat{c}_i$ as the input of the individual utility Q_i, which is used for the agent to make decisions. For the reinforcement learning part, we follow the design of the classical value decomposition method, using the TD loss $\mathcal{L}_{\text{TD}}$ from Eq. (2) to train the individual policies of agents.

Overall, the entire algorithm framework includes the following optimization components: the TD loss $\mathcal{L}_{\text{TD}}$ for reinforcement learning, the consensus loss $\mathcal{L}_{\text{Cons}}$ for the consensus generation part, the mutual information optimization term $\mathcal{L}_{\text{Msg}}$ for the message generation part, and the cross-entropy loss $\mathcal{L}_g$ for the gate mechanism learning. Therefore, the total loss function for the entire algorithm framework is:

$$\mathcal{L}_{tot} = \mathcal{L}_{\text{TD}} + \mathcal{L}_{\text{Cons}} + \mathcal{L}_{\text{Msg}} + \mathcal{L}_g. \tag{16}$$

Experiments

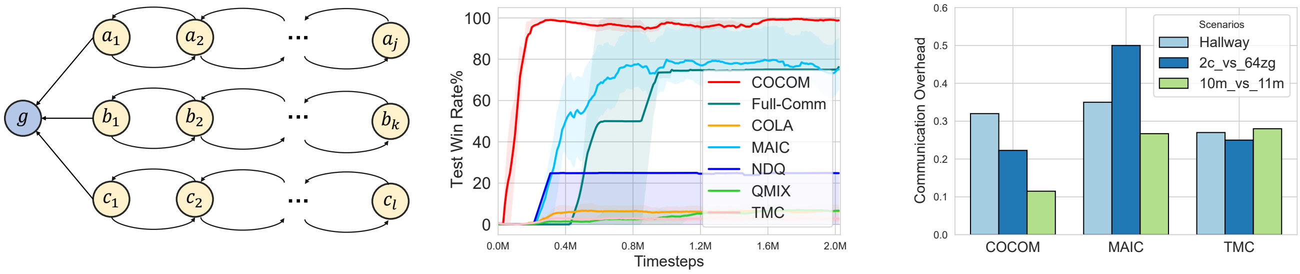
In this chapter, we address the following key questions through a series of detailed experiments: (i) Can the COCOM algorithm effectively reduce communication costs while maintaining performance? (ii) What communication patterns does COCOM learn? (iii) Can the COCOM algorithm be combined with different value decomposition baselines to enhance their performance?

We compare the COCOM algorithm with a range of baseline algorithms, including the classic value decomposition algorithms QMIX (Rashid et al. 2018), VDN (Sunehag et al. 2018) which is communication-free, the COLA (Xu et al. 2023) algorithm that only learns implicit consensus, and the Full-Comm (Guan et al. 2022) and TarMAC (Das et al. 2019) algorithms, which do not consider communication costs. Additionally, we compare against some state-of-the-art communication algorithms that also reduce communication costs by filtering messages, such as NDQ (Wang et al. 2020), MAIC (Yuan et al. 2022), and TMC (Zhang, Zhang, and Lin 2020). We adopt the fine-tuned parameters in PyMARL2 (Hu et al. 2023) to ensure the performance of baseline algorithms.

We conducted our experiments in Hallway (Wang et al. 2020), StarCraft II Multi-Agent Challenge benchmark (SMAC) (Samvelyan et al. 2019), and Google Research Football (GRF) (Kurach et al. 2020). All curves are presented with average performance and 25$\sim$75% deviation over five random seeds, with the solid lines representing the average win rates.

Performance on Hallway

The Hallway task (Yuan et al. 2022) is a multi-agent cooperative environment where three agents are randomly initialized on three routes. The lengths of the three routes are j,



(a) Hallway Task

(b) Performance comparison on Hallway

(c) Communication Overhead

Figure 2: (a) Map of Hallway task. (b) Average test win rate on Hallway task. (c) The communication overhead comparison.

k, and l, respectively. Each agent can only observe its current position and choose to move left, right, or stay still. The episode ends when any agent reaches the target, but a reward of +10 is given only if all agents reach the goal point simultaneously. We set j=2, k=6, and l=10. In this task, since agents cannot observe the positions of their teammates, learning an effective communication module is crucial. The Hallway task is shown in Figure 2(a).

Figure 2(b) shows the performance of different baselines on the Hallway task. The communication-free algorithm QMIX performs the worst in this scenario. The COLA algorithm, which builds consensus based on local observations, also performs poorly. Although it can form a consensus to move toward the target, the lack of explicit communication prevents agents from knowing their teammates' positions, making it difficult to reach the goal simultaneously. The Full-Comm algorithm performs well overall. However, because the Full-Comm method does not filter messages, agents can be overwhelmed by redundant information, leading to inefficient learning. Other communication algorithms excessively filter messages but fail to provide alternative guiding information when messages are suppressed, leading to instability in learning. The COCOM algorithm, however, can still make decisions using consensus information without communication and only communicates at critical time steps, ensuring both efficiency and low communication costs.

Visualization on Hallway Task

To investigate whether the COCOM method learns an effective communication strategy, we visualized an entire episode in the Hallway task, as shown in Figure 3. The upper part of Figure 3 illustrates the positions of the agents at each time step, while the lower part displays whether the message gates of the three agents are open at each time step. The three agents, a, b, and c, are initialized at positions a_2, b_3, and c_4, respectively. Guided by consensus, all agents can move synchronously toward the target without explicit communication during the initial phase. The agents only begin communicating upon reaching the vicinity of the target. At the last timestep, through explicit communication, all agents become aware that their teammates can approach the target, so they move toward the target simultaneously and successfully complete the task.

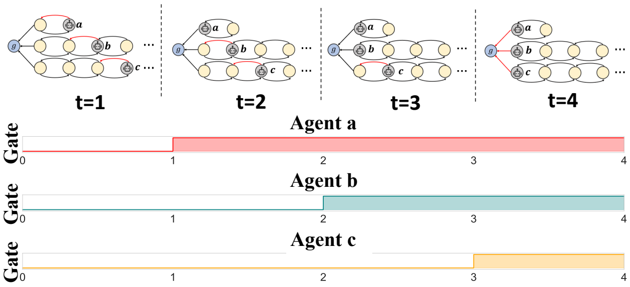


Figure 3: Case study and visualization on Hallway.

Performance on SMAC

The SMAC is a multi-agent environment based on the real-time strategy game Starcraft, where there are numerous micromanagement tasks of various difficulty levels with locally observable properties. We selected six scenarios of SMAC including four easy scenarios *2s3z*, *3s5z*, *1c3s5z*, *10m_vs_11m*, one hard scenarios *2c_vs_64zg*, and one super hard scenarios *MMM2* for evaluation on the StarCraft II's version SC2.4.10.

The results on SMAC are shown in Figure 4. Unlike the Hallway scenario, in SMAC, agents can observe some of their teammates' information. Methods like Full-Comm that send unfiltered messages may distract the agents and degrade performance in SMAC. In contrast, the COCOM algorithm leverages consensus to refine and filter messages, leading to superior performance across different SMAC maps compared to existing communication and consensus methods. This demonstrates that the consensus module and communication module can effectively promote each other.

Communciation Overhead

We next examine the communication costs of various algorithms across different scenarios, including COCOM and two communication algorithms with message pruning, MAIC and TMC. To measure communication cost, we use

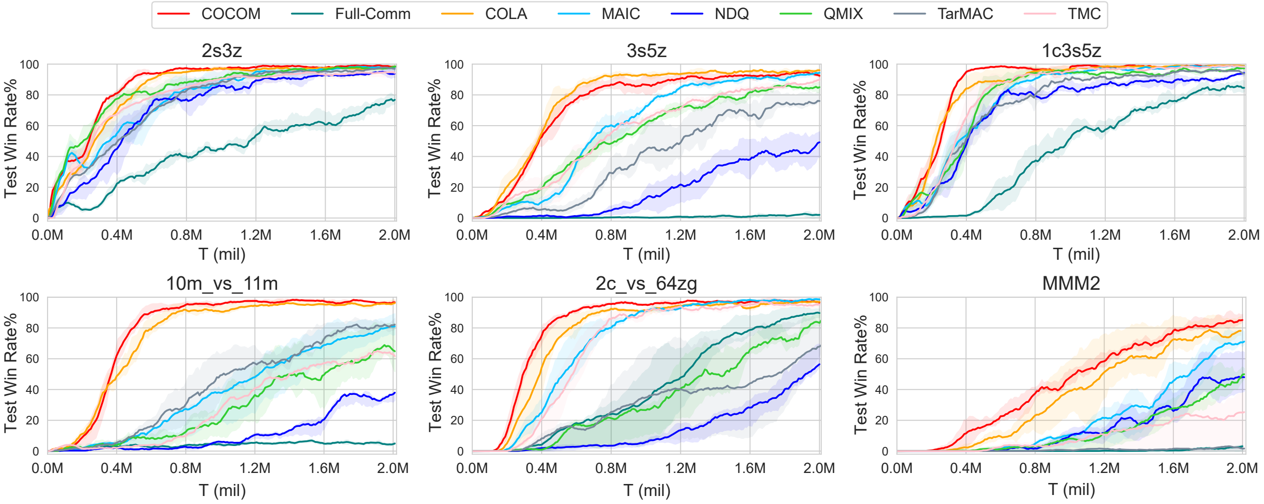
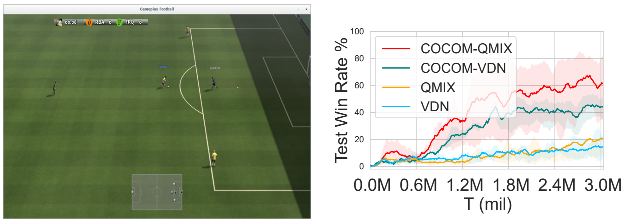


Figure 4: Performance comparison on SMAC

a method similar to previous work (Zhang, Zhang, and Lin 2020, 2019), calculating the average communication pairs throughout the entire episode: $\frac{\sum_{t=0}^{T} x_t}{ZT}$, where x_t is the number of agent pairs communicating at each time step, and Z is the total number of agent pairs. As shown in Figure 2(c), the COCOM algorithm demonstrates adaptive communication characteristics. It effectively reduces communication costs based on the requirements of different scenarios. In the SMAC scenario (*2c_vs_64zg*, *10m_vs_11m*), where communication demands are lower, COCOM achieves the lowest communication costs among all algorithms. In the Hallway scenario, where agents cannot observe their teammates, COCOM maintains communication costs at a level that ensures task completion. The MAIC algorithm has the highest overhead due to its lack of signals to replace communication messages. The TMC algorithm compresses communication by similar proportions across different scenarios, leading to excessive communication reduction in the Hallway scenario, making it difficult to complete the task.

Performance on GRF

The Google Research Football environment (Kurach et al. 2020) is a multi-agent platform designed for training agents to play soccer, as illustrated in Figure 5(a). In this environment, multiple agents must collaborate to achieve various sub-tasks. We conduct experiments in the GRF to evaluate the performance of the COCOM combined with different value decomposition frameworks. We test on the *academy_3_vs_1_with_keeper* scenario, using two baseline algorithms, VDN and QMIX, and examining the results of integrating COCOM with these algorithms, termed COCOM-VDN and COCOM-QMIX, respectively. The experimental results, shown in Figure 5(b), show that combining COCOM with different value decomposition algorithms significantly improves the performance of the original algorithms, demonstrating that COCOM can be effectively extended to various value decomposition frameworks.



(a)

(b)

Figure 5: (a) Screenshot of GRF. (b) Results on GRF

Conclusion

This paper introduces COCOM, a novel and efficient communication method that combines explicit communication with implicit consensus learning. In COCOM, agents infer a shared consensus based on their local observations, which can guide decision-making and reduce instability caused by communication interruptions. Additionally, the consensus facilitates the generation of more concise messages. The COCOM algorithm performs well across various scenarios, significantly reducing communication costs. COCOM can be integrated with different value decomposition variants to enhance the performance of existing algorithms. Currently, COCOM broadcasts the message to all teammates. The future work will explore incorporating customized message generation to broaden its applicability. As the multi-agent research community increasingly emphasizes algorithmic practicality, COCOM's efficiency and low communication costs will be highly advantageous for future research.

Acknowledgments

This project was supported by Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDA27050100.

References

Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; Debiak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; Józefowicz, R.; Gray, S.; Olsson, C.; Pachocki, J.; Petrov, M.; de Oliveira Pinto, H. P.; Raiman, J.; Salimans, T.; Schlatter, J.; Schneider, J.; Sidor, S.; Sutskever, I.; Tang, J.; Wolski, F.; and Zhang, S. 2019. Dota 2 with Large Scale Deep Reinforcement Learning. *CoRR*, abs/1912.06680.

Bernstein, D. S.; Givan, R.; Immerman, N.; and Zilberstein, S. 2002. The Complexity of Decentralized Control of Markov Decision Processes. *Math. Oper. Res.*, 27(4): 819–840.

Cho, K.; van Merrienboer, B.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Moschitti, A.; Pang, B.; and Daelemans, W., eds., *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 1724–1734. ACL.

Das, A.; Gervet, T.; Romoff, J.; Batra, D.; Parikh, D.; Rabbat, M.; and Pineau, J. 2019. TarMAC: Targeted Multi-Agent Communication. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 1538–1546. PMLR.

de Witt, C. S.; Foerster, J. N.; Farquhar, G.; Torr, P. H. S.; Boehmer, W.; and Whiteson, S. 2019. Multi-Agent Common Knowledge Reinforcement Learning. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 9924–9935.

Ding, Z.; Huang, T.; and Lu, Z. 2020. Learning individually inferred communication for multi-agent cooperation. *Advances in Neural Information Processing Systems*, 33: 22069–22079.

Foerster, J. N.; Assael, Y. M.; de Freitas, N.; and Whiteson, S. 2016. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In Lee, D. D.; Sugiyama, M.; von Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2137–2145.

Foerster, J. N.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2018. Counterfactual Multi-Agent Policy Gradients. In McIlraith, S. A.; and Weinberger, K. Q., eds., *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2974–2982. AAAI Press.

Guan, C.; Chen, F.; Yuan, L.; Wang, C.; Yin, H.; Zhang, Z.; and Yu, Y. 2022. Efficient Multi-agent Communication via Self-supervised Information Aggregation. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Horowitz, I. L. 1962. Consensus, conflict and cooperation: a sociological inventory. *Social Forces*, 41(2): 177–188.

Hu, J.; Wang, S.; Jiang, S.; and Wang, M. 2023. Rethinking the Implementation Tricks and Monotonicity Constraint in Cooperative Multi-agent Reinforcement Learning. In *The Second Blogpost Track at ICLR 2023*.

Jiang, J.; Dun, C.; Huang, T.; and Lu, Z. 2020. Graph Convolutional Reinforcement Learning. In *International Conference on Learning Representations*.

Jiang, J.; and Lu, Z. 2018. Learning Attentional Communication for Multi-Agent Cooperation. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 7265–7275.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In Bengio, Y.; and LeCun, Y., eds., *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Kurach, K.; Raichuk, A.; Stanczyk, P.; Zajac, M.; Bachem, O.; Espeholt, L.; Riquelme, C.; Vincent, D.; Michalski, M.; Bousquet, O.; and Gelly, S. 2020. Google Research Football: A Novel Reinforcement Learning Environment. 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*, 4501–4510. AAAI Press.

Li, D.; Lou, N.; Zhang, B.; Xu, Z.; and Fan, G. 2024a. Adaptive Parameter Sharing for Multi-Agent Reinforcement Learning. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6035–6039.

Li, D.; Xu, Z.; Zhang, B.; Zhou, G.; Zhang, Z.; and Fan, G. 2024b. From Explicit Communication to Tacit Cooperation: A Novel Paradigm for Cooperative MARL. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, 2360–2362. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400704864.

Li, X.; and Zhang, J. 2024. Context-aware Communication for Multi-agent Reinforcement Learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, 1156–1164. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400704864.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; and Mordatch, I. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In Guyon, I.; von

Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 6379–6390.

Mao, H.; Liu, W.; Hao, J.; Luo, J.; Li, D.; Zhang, Z.; Wang, J.; and Xiao, Z. 2020a. Neighborhood Cognition Consistent Multi-Agent Reinforcement Learning. 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*, 7219–7226. AAAI Press.

Mao, H.; Zhang, Z.; Xiao, Z.; Gong, Z.; and Ni, Y. 2020b. Learning Agent Communication under Limited Bandwidth by Message Pruning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04): 5142–5149.

Pagello, E.; D'Angelo, A.; Montesello, F.; Garelli, F.; and Ferrari, C. 1999. Cooperative behaviors in multi-robot systems through implicit communication. *Robotics and Autonomous Systems*, 29(1): 65–77.

Parikh, R.; and Krasucki, P. 1990. Communication, consensus, and knowledge. *Journal of Economic Theory*, 52(1): 178–189.

Rashid, T.; Samvelyan, M.; de Witt, C. S.; Farquhar, G.; Foerster, J. N.; and Whiteson, S. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In Dy, J. G.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 4292–4301. PMLR.

Roth, M.; Simmons, R.; and Veloso, M. 2006. What to communicate? Execution-time decision in multi-agent POMDPs. In *Distributed autonomous robotic systems 7*, 177–186. Springer.

Ruan, J.; Hao, X.; Li, D.; and Mao, H. 2023. *Learning to Collaborate by Grouping: A Consensus-Oriented Strategy for Multi-Agent Reinforcement Learning*. ISBN 9781643684369.

Samvelyan, M.; Rashid, T.; de Witt, C. S.; Farquhar, G.; Nardelli, N.; Rudner, T. G. J.; Hung, C.; Torr, P. H. S.; Foerster, J. N.; and Whiteson, S. 2019. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043.

Shao, J.; Zhang, H.; Qu, Y.; Liu, C.; He, S.; Jiang, Y.; and Ji, X. 2023. Complementary attention for multi-agent reinforcement learning. In *International Conference on Machine Learning*, 30776–30793. PMLR.

Sukhbaatar, S.; Szlam, A.; and Fergus, R. 2016. Learning Multiagent Communication with Backpropagation. In Lee, D. D.; Sugiyama, M.; von Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2244–2252.

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J. Z.; Tuyls, K.; and Graepel, T. 2018. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, 2085–2087. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; Oh, J.; Horgan, D.; Kroiss, M.; Danihelka, I.; Huang, A.; Sifre, L.; Cai, T.; Agapiou, J. P.; Jaderberg, M.; Vezhnevets, A. S.; Leblond, R.; Pohlen, T.; Dalibard, V.; Budden, D.; Sulsky, Y.; Molloy, J.; Paine, T. L.; Gülçehre, Ç.; Wang, Z.; Pfaff, T.; Wu, Y.; Ring, R.; Yogatama, D.; Wünsch, D.; McKinney, K.; Smith, O.; Schaul, T.; Lillicrap, T. P.; Kavukcuoglu, K.; Hassabis, D.; Apps, C.; and Silver, D. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nat.*, 575(7782): 350–354.

Wang, J.; Xu, W.; Gu, Y.; Song, W.; and Green, T. C. 2021. Multi-agent reinforcement learning for active voltage control on power distribution networks. *Advances in Neural Information Processing Systems*, 34: 3271–3284.

Wang, T.; Wang, J.; Zheng, C.; and Zhang, C. 2020. Learning Nearly Decomposable Value Functions Via Communication Minimization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xu, Z.; Zhang, B.; Li, D.; Zhang, Z.; Zhou, G.; Chen, H.; and Fan, G. 2023. Consensus learning for cooperative multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 11726–11734.

Yuan, L.; Wang, J.; Zhang, F.; Wang, C.; Zhang, Z.; Yu, Y.; and Zhang, C. 2022. Multi-Agent Incentive Communication via Decentralized Teammate Modeling. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, 9466–9474. AAAI Press.

Zhang, S. Q.; Zhang, Q.; and Lin, J. 2019. Efficient communication in multi-agent reinforcement learning via variance based control. *Advances in Neural Information Processing Systems*, 32.

Zhang, S. Q.; Zhang, Q.; and Lin, J. 2020. Succinct and robust multi-agent communication with temporal message control. *Advances in Neural Information Processing Systems*, 33: 17271–17282.