Multi-Agent Incentive Communication via Decentralized Teammate Modeling*

**Lei Yuan†,[1,3] Jianhao Wang†,[2] Fuxiang Zhang†,[1] Chenghe Wang,[1]
Zongzhang Zhang,[1] Yang Yu,[1,3,4] Chongjie Zhang [2]**

[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[2]Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, China
[3]Polixir Technologies, Nanjing 210000, China
[4]Peng Cheng Laboratory, Shenzhen 518055, China
yuanl@lamda.nju.edu.cn, wjh19@mails.tsinghua.edu.cn, zhangfx@lamda.nju.edu.cn, wangch@lamda.nju.edu.cn,
zzzhang@nju.edu.cn, yuy@nju.edu.cn, chongjie@tsinghua.edu.cn

Abstract

Effective communication can improve coordination in cooperative multi-agent reinforcement learning (MARL). One popular communication scheme is exchanging agents' local observations or latent embeddings and using them to augment individual local policy input. Such a communication paradigm can reduce uncertainty for local decision-making and induce implicit coordination. However, it enlarges agents' local policy spaces and increases learning complexity, leading to poor coordination in complex settings. To handle this limitation, this paper proposes a novel framework named *Multi-Agent Incentive Communication* (MAIC) that allows each agent to learn to generate incentive messages and bias other agents' value functions directly, resulting in effective explicit coordination. Our method firstly learns targeted teammate models, with which each agent can anticipate the teammate's action selection and generate tailored messages to specific agents. We further introduce a novel regularization to leverage interaction sparsity and improve communication efficiency. MAIC is agnostic to specific MARL algorithms and can be flexibly integrated with different value function factorization methods. Empirical results demonstrate that our method significantly outperforms baselines and achieves excellent performance on multiple cooperative MARL tasks.

Introduction

Cooperative multi-agent reinforcement learning (MARL) has attracted popular attention (Hernandez-Leal, Kartal, and Taylor 2019; Du and Ding 2021), showing a promise in many domains like autonomous vehicle teams (Zhou et al. 2020), sensor networks (Zhang and Lesser 2011) and intelligent warehouse systems (Nowé, Vrancx, and De Hauwere 2012). To avoid non-stationarity and enable scalability, a popular MARL paradigm, called *Centralized Training and Decentralized Execution* (CTDE) (Kraemer and Banerjee 2016; Lyu et al. 2021), has recently been adopted, where agents' policies are learned in a centralized way and executed in a decentralized manner. Many CTDE learning approaches have been proposed, including both policy gradient methods (Lowe et al. 2017; Foerster et al. 2018; Wang et al. 2020a, 2021b; Zhang et al. 2021), and value-based methods (Rashid et al. 2018; Wang et al. 2020b; Son et al. 2019; Wang et al. 2021a, 2020c; Cao et al. 2021), which show state-of-the-art performance in challenging tasks (e.g., the StarCraft II micromanagement benchmark). Despite its recent success, fully decentralization policy execution may not be effective in many applications, especially when agents have partial observability and the environment is stochastic, where an agent's uncertainty of other agent's states and actions can be exacerbated during decentralized sequential execution, which results in catastrophic miscoordination and sub-optimal policies.

An effective mechanism to address the mentioned challenges is to enable communication among agents. One popular communication scheme has been adopted to exchange messages about agents' local observations (Sukhbaatar, Szlam, and Fergus 2016; Singh, Jain, and Sukhbaatar 2019; Ding, Huang, and Lu 2020) or corresponding embeddings (Das et al. 2019; Zhang, Zhang, and Lin 2019; Wang et al. 2020d; Mao et al. 2020b). Such messages are then used to augment individual local observations for learning policies and selecting actions. This informative communication scheme can reduce uncertainty for individual local decision-making and induce implicit coordination. However, such a communication mechanism enlarges agents' local policy spaces and increases learning complexity, leading to poor performance in some complex scenery. Furthermore, this scheme deviates a common way humans communicate, as humans would often offer helpful and tailored suggestions based on their knowledge and beliefs instead of simply providing their own information (Stenning, Lascarides, and Calder 2006).

Towards realizing efficient and human-like teamwork, this paper investigates a new communication scheme that allows agents to exchange persuasive messages to coordinate their decision-making explicitly. We propose a novel MARL framework called *Multi-Agent Incentive Communication* (MAIC), where each agent learns to generate incentive messages and uses the messages to bias other agents' value functions directly for effective coordination. Inspired by recent opponent modeling works (Albrecht and Stone 2018),

*Corresponding authors: Zongzhang Zhang and Chongjie Zhang.
†These authors contributed equally.

the MAIC agent learns a targeted teammate model for every other agent from its local observation. This teammate model, which maximizes a mutual information regularizer to associate learned models with opponent intention anticipation, can help each agent dynamically generate tailored incentive messages to specific teammates without enlarging policy spaces. We utilize the teammate model to design a novel message generator to produce messages, along with distinguished communication weights. The emergence of communication weights can tackle interaction sparsity, a commonly existing structure in multi-agent systems, by pruning messages with minor communication weights. Therefore, we explicitly introduce a sparse regularizer at communication weights to distinguish valuable agents to communicate with. In this way, our method provides a tailored and sparse communication paradigm by teammate modeling with incentive interaction, promoting the coordination ability in cooperative tasks. We extensively evaluate MAIC on diverse MARL benchmarks, including level-based foraging (Papoudakis et al. 2021), Hallway (Wang et al. 2020d), and StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al. 2019). Empirical results show that MAIC can generate dynamic and individualized messages for specific teammates to achieve outstanding performance and effective coordination.

Related Work

Recently, communication in MARL has attracted widespread attention (Giles and Jim 2002; Foerster et al. 2016; Hernandez-Leal, Kartal, and Taylor 2019; Lazaridou and Baroni 2020; Xue et al. 2021; Du et al. 2021). DAIL (Foerster et al. 2016) is a simple communication mechanism where agents broadcast messages to all teammates that allow the gradient to flow among agents for end-to-end training with reinforcement learning. Nevertheless, this method is limited to discrete messages. CommNet (Sukhbaatar, Szlam, and Fergus 2016) proposes an efficient centralized communication structure, where the outputs of the hidden layers from all the agents are collected and averaged to augment local observation. It leads to information semantics loss and limits its performance in complex scenarios. VBC (Zhang, Zhang, and Lin 2019) adds a variance-based regularizer to eliminate the noisy component in the messages, realizing lower communication overhead and better performance than other MARL communication methods. However, VBC needs to send the local hidden information to teammates, which causes bandwidth wasting inevitably. To alleviate the burden of local policy caused by flooding messages, IC3Net (Singh, Jain, and Sukhbaatar 2019), Gated-ACML (Mao et al. 2020a), and I2C (Ding, Huang, and Lu 2020) learn a gate mechanism to decide whom to communicate with. These methods work well in applications such as traffic junction, packet routing, and MPE (Lowe et al. 2017). Nevertheless, they use their own observations (or encodings of their own observations) as messages to augment the local policy. This paradigm makes the local policy complex and hinders the learning process.

Research on learning targeted and efficient communication has made some progress recently. TarMAC (Das et al. 2019) sends coordinated messages with additional signatures and received messages from different agents derive the attention outputs to enlarge the policy. TarMAC needs to broadcast an agent's information to all teammates, leading to severe bandwidth waste. DAACMP (Mao et al. 2020b) adds a double attention mechanism in the actor-critic framework based on MADDPG (Lowe et al. 2017), showing attention can significantly improve the performance of multi-agent systems. However, DAACMP also lacks knowledge about specific teammates. NDQ (Wang et al. 2020d) utilizes a minimized communication paradigm to alter value functions based on two different information-theory-based regularizers. Compared with our approach, the regularized message enlarges local policy space, impairing the learning process, and the uncertainty of messages limits its performance in complex environments. TMC (Zhang, Zhang, and Lin 2020) realizes succinct and robust communication by applying smoothing and action selection regularizers. TMC processes received messages by adding to individual value functions as incentives, which is an efficient and promising way. However, the lack of teammate modeling in TMC also results in the broadcast paradigm and hinders its empirical performance.

To our knowledge, none of the existing MARL communication methods considers generating incentive messages through teammate modeling. Therefore, agents may not only be confused by redundant information in broadcast-like communication but also suffer from learning policies from a larger policy space augmented by messages. MAIC enables agents to hold specific teammate models, generate tailored incentives for others, and dynamically prune useless communication, realizing effective and sparse message exchange.

Background

This paper considers a fully cooperative multi-agent task with partial observations, which can be modeled as a Dec-POMDP (Oliehoek and Amato 2016). A Dec-POMDP can be defined by a tuple $\mathcal{M} = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}, P, \Omega, O, R, \gamma \rangle$, where $\mathcal{N} = \{1, \ldots, n\}$ is the set of agents, $\mathcal{S}$ is the set of global states, $\mathcal{A}$ is the set of actions, Ω is the set of observations, and $\gamma \in [0, 1)$ stands for the discounted factor. At each time step, every agent $i \in \mathcal{N}$ can only acquire the observation $o_i \in \Omega$ drawn from the observation function $O(s, i)$ where $s \in \mathcal{S}$, and then choose the action $a_i \in \mathcal{A}$. The joint action $\boldsymbol{a} = \langle a_1, \ldots, a_n \rangle$ leads to next state $s' \sim P(s' \mid s, \boldsymbol{a})$ and the global reward $R(s, \boldsymbol{a})$. The formal objective is to find a joint policy $\boldsymbol{\pi}(\boldsymbol{\tau}, \boldsymbol{a})$ to maximize the global value function $Q_{\text{tot}}^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a}) = \mathbb{E}_{s,\boldsymbol{a}} \left[\sum_{t=0}^{\infty} \gamma^t R(s, \boldsymbol{a}) \mid s_0 = s, \boldsymbol{a_0} = \boldsymbol{a}, \boldsymbol{\pi}\right]$. Here, $\boldsymbol{\tau} = \langle \tau_1, \ldots, \tau_n \rangle$, and τ_i represents the history $(o_i^1, a_i^1, \ldots, o_i^{t-1}, a_i^{t-1}, o_i^t)$ of agent i at current timestep t.

Q-learning (Sutton and Barto 2018) is a widely-used algorithm to find the optimal join action-value function $Q^*(s, \boldsymbol{a}) = r(s, \boldsymbol{a}) + \gamma \mathbb{E}_{s'} \left[\max_{\boldsymbol{a}'} Q(s', \boldsymbol{a}')\right]$. Deep Q-learning (Mnih et al. 2015) represents the action-value function $Q^*(s, \boldsymbol{a})$ with a deep neural network $Q(\boldsymbol{\tau}, \boldsymbol{a}; \boldsymbol{\theta})$ parameterized by $\boldsymbol{\theta}$. In the training phase, deep Q-learning uses a replay memory $\mathcal{D}$ to store the transition tuple $\langle \boldsymbol{\tau}, \boldsymbol{a}, r, \boldsymbol{\tau}' \rangle$. We use $Q(\boldsymbol{\tau}, \boldsymbol{a}; \boldsymbol{\theta})$ to approximate $Q(s, \boldsymbol{a}; \boldsymbol{\theta})$ to relieve the partial observable problem. Thus, the parameters $\boldsymbol{\theta}$ are learnt by minimizing the expected TD error:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{\tau}, \boldsymbol{a}, r, \boldsymbol{\tau}') \in \mathcal{D}} \left[\left(r + \gamma V \left(\boldsymbol{\tau}'; \boldsymbol{\theta}^- \right) - Q(\boldsymbol{\tau}, \boldsymbol{a}; \boldsymbol{\theta}) \right)^2 \right],$$

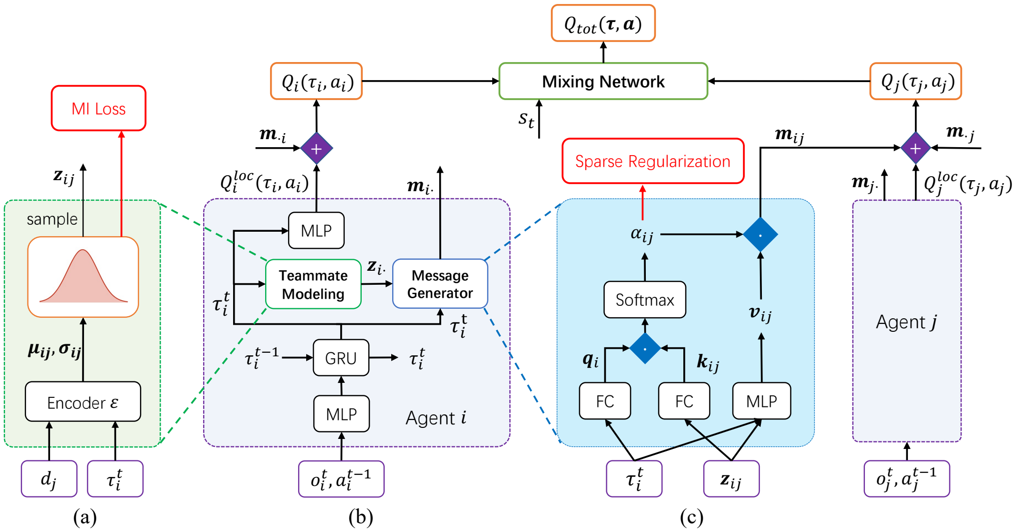


Figure 1: Schematics of MAIC. (a) Decentralized teammate modeling. (b) Network for agent i. (c) The message generator.

where $V(\tau'; \boldsymbol{\theta}^-) = \max_{\boldsymbol{a}'} Q(\boldsymbol{\tau}', \boldsymbol{a}'; \boldsymbol{\theta}^-)$ is the expected future return of the TD target and $\boldsymbol{\theta}^-$ are parameters of the target network, which will be periodically updated with $\boldsymbol{\theta}$.

Method

In this section, we will describe the detailed design of MAIC (Figure 1). For decentralized decision making, each MAIC agent takes its obtained observation and last action as input and feeds them into a GRU cell (Cho et al. 2014) to get a representation of historical information. We further use this historical representation to generate local Q-values $Q_i^{\text{loc}}(\tau_i, a_i)$ by a multi-layer perceptron (MLP). The local network of each MAIC agent (Figure 1(b)) also contains a teammate modeling network (Figure 1(a)) and a tailored message generator (Figure 1(c)). The MAIC agent utilizes sampled representation from teammate models to generate sparse communication weights and tailored message contents. Processed messages are then fed to other agents' policies in an incentive way, resulting in efficient and sparse communication.

Decentralized Teammate Modeling

To guide the intentions for specific agents, MAIC learns targeted teammates models, which can infer the action selection of these agents in a partially observable way (Figure 1(a)). We leverage this local information τ_i and the specific teammate ID d_j to obtain a teammate model from agent i to agent j. The teammate model is represented by a multivariate Gaussian distribution whose parameters $(\boldsymbol{\mu}_{ij}, \boldsymbol{\sigma}_{ij})$ are computed by an encoder $\mathcal{E}(\tau_i, d_j)$ with multiple fully connected layers. A valid teammate representation $\boldsymbol{z}_{ij}$ for agent j is then sampled from the teammate model $N(\boldsymbol{\mu}_{ij}, \boldsymbol{\sigma}_{ij}^2)$ to guide the message generation process of agent i.

As the simple MLP cannot guarantee the anticipation of teammate models, we introduce an explicit regularization to guide the teammate modeling. We hope the learned teammate models should be responsive to the action selection of every specific teammate, as the selected actions can intuitively exhibit the coordination relation among agents. We optimize the teammate models by maximizing the mutual information (MI) between the action a_j taken by agent j and the random variable $\boldsymbol{z}_{ij}$ of the teammate model distribution conditioned on agent i's local trajectory τ_i and the specific ID of teammate j, d_j. We express the MI term via the entropy $H(\boldsymbol{z}_{ij} \mid \tau_i, d_j)$ and the conditional entropy $H(\boldsymbol{z}_{ij} \mid \tau_i, a_j, d_j)$:

$$I(\boldsymbol{z}_{ij}, a_j \mid \tau_i, d_j) = H(\boldsymbol{z}_{ij} \mid \tau_i, d_j) - H(\boldsymbol{z}_{ij} \mid \tau_i, a_j, d_j).$$

If the knowledge of $H(\boldsymbol{z}_{ij} \mid \tau_i, a_j, d_j)$ does not provide any information about $H(\boldsymbol{z}_{ij} \mid \tau_i, d_j)$, the conditional entropy would reduce to the unconditional entropy, i.e., $H(\boldsymbol{z}_{ij} \mid \tau_i, d_j) = H(\boldsymbol{z}_{ij} \mid \tau_i, a_j, d_j)$ and the mutual information would become zero. Maximizing MI is equivalent to minimizing the uncertainty about the learned teammate model conditioned on the agent's local information, leading agent i to acquire a powerful teammate representation $\boldsymbol{z}_{ij}$ when modeling agent j.

Unfortunately, it is difficult to compute the conditional distribution directly. Inspired by the information bottleneck (Alemi et al. 2017), we use $q_\xi(\boldsymbol{z}_{ij} \mid \tau_i, a_j, d_j)$ as a variational distribution to approximate the conditional distribution $p(\boldsymbol{z}_{ij} \mid \tau_i, d_j)$, and derive a lower bound for the MI term:

$$\begin{aligned} & I(\boldsymbol{z}_{ij}; a_j \mid \tau_i, d_j) \geq \\ & \quad \mathbb{E}_{\mathcal{D}}\left[-D_{\text{KL}}\left(p(\boldsymbol{z}_{ij} \mid \tau_i, d_j) \,\|\, q_\xi(\boldsymbol{z}_{ij} \mid \tau_i, a_j, d_j)\right)\right], \end{aligned}$$

where the variables of distribution p and q_ξ are sampled from the experience replay buffer $\mathcal{D}$ and D_{KL} denotes the Kullback-Leibler divergence. We can rewrite the lower bound and derive the teammate modeling loss function:

$$\begin{aligned} & \mathcal{L}_m(\boldsymbol{\theta}_m) = \\ & \sum_{i \neq j} \mathbb{E}_{\mathcal{D}}\left[D_{\text{KL}}(p(\boldsymbol{z}_{ij} \mid \tau_i, d_j) \,\|\, q_\xi(\boldsymbol{z}_{ij} \mid \tau_i, a_j, d_j))\right], \end{aligned} \quad (1)$$

where $\boldsymbol{\theta}_m$ defines all parameters including parameters of the teammate modeling encoder and the variational distribution.

Multi-Agent Incentive Communication

As the MAIC agent can extract targeted teammate information from learned teammate models, this representation can be utilized to generate tailored messages for different agents (Figure 1(c)). Specifically, MAIC maintains a local Q-network to compute the local Q-values $Q_i^{\mathrm{loc}}(\tau_i, a_i)$ for each action $a_i \in \mathcal{A}$. The message generator contains an MLP $\boldsymbol{v}_{ij} = f_m(\tau_i, \boldsymbol{z}_{ij})$, whose input includes the local observation history τ_i of agent i as well as the teammate representation $\boldsymbol{z}_{ij}$ for agent $j \in \{1, \cdots, i-1, i+1, \cdots, n\}$. The output $\boldsymbol{v}_{ij}$ has the same dimension as the local Q network, which is the dimension of the action space.

The teammate representation can also help communicate in a more targeted way, as unnecessary information may confuse the receiver and impair coordination. An MAIC agent will calculate communication weights for all other agents by taking the teammate representation as input. To better combine the representation with the agent's own information, we apply an attention-like mechanism (Vaswani et al. 2017) to compute the query $\boldsymbol{q}_i$ from the agent's trajectory τ_i and the key $\boldsymbol{k}_{ij}$ from the representation of teammate j. Both $\boldsymbol{q}_i$ and $\boldsymbol{k}_{ij}$ are computed by simple linear functions. The communication weight α_{ij}, which defines the weight to agent j for agent i's communication, is normalized to make all α_{ij} from agent i form a categorical distribution:

$$\alpha_{ij} = \frac{1}{\eta} \exp(-\lambda \boldsymbol{q}_i^{\mathrm{T}} \boldsymbol{k}_{ij}), \tag{2}$$

where $\eta = \sum_{m \neq i} \exp(-\lambda \boldsymbol{q}_i^{\mathrm{T}} \boldsymbol{k}_{im})$ and $\lambda \in \mathbb{R}^+$ is the temperature parameter to scale the magnitude of input. The final message from agent i to agent j can be calculated by a product $\boldsymbol{m}_{ij} = \alpha_{ij} \boldsymbol{v}_{ij}$. As our message is already represented in an effective and targeted way, we utilize the final message to bias the other agent's Q-values as an incentive, because this approach would not explicitly enlarge the policy space of every agent. Specifically,

$$Q_i(\tau_i, \cdot) = Q_i^{\mathrm{loc}}(\tau_i, \cdot) + \sum\nolimits_{j \neq i} \boldsymbol{m}_{ji}, \tag{3}$$

where $Q_i(\tau_i, \cdot)$ and $Q_i^{\mathrm{loc}}(\tau_i, \cdot)$ denote the corresponding Q-value vector for every action a_i.

Sparse Communication Among Agents Even though we can generate different communication weights for different teammates, the neural network itself cannot force agents to produce sparse weights, resulting in uniform communication weights for different agents. To learn sparse yet effective communication, we further introduce a sparsity regularization, which optimizes the entropy of the category distribution formed by communication weights:

$$\mathcal{L}_c(\boldsymbol{\theta}_c) = \sum\nolimits_{j=1}^{n} \mathcal{H}(\alpha_{ij}) = -\sum\nolimits_{j=1}^{n} \alpha_{ij} \log \alpha_{ij}, \tag{4}$$

where $\boldsymbol{\theta}_c$ are parameters of the message generator. To minimize this entropy loss, we can acquire communication weights with lower uncertainty. Furthermore, it is available to cut useless communication links and reduce redundant information through the communication weight. As the communication weight should be less than 1 and their summation equals 1, we can remove the communication link from agent i to agent j when $\alpha_{ij} < \frac{\delta}{n}$, where $\delta \in (0, 1]$ represents the message sparsity threshold.

Overall Optimization Objective

As the MAIC framework is implemented with the CTDE paradigm, all parameters in it are updated by the standard TD loss from the global Q-values Q_{tot}, which are the output of any mixing network such as VDN (Sunehag et al. 2018), QMIX (Rashid et al. 2018), and QPLEX (Wang et al. 2021a):

$$\mathcal{L}_{\mathrm{TD}}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{\tau}, \boldsymbol{a}, r, \boldsymbol{\tau}') \sim \mathcal{D}} \left[(y - Q_{\mathrm{tot}}(\boldsymbol{\tau}, \boldsymbol{a}; \boldsymbol{\theta}))^2 \right], \tag{5}$$

where $y = r + \max_{\boldsymbol{a}'} Q_{\mathrm{tot}}(\boldsymbol{\tau}', \boldsymbol{a}'; \boldsymbol{\theta}^-)$ is the target, and $\boldsymbol{\theta}^-$ are parameters of the periodically updated target network. Together with the mentioned teammate modeling loss and sparsity regularization, the learning objective of MAIC is:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\mathrm{TD}}(\boldsymbol{\theta}) + \lambda_m \sum_{i=1}^{n} \mathcal{L}_m(\boldsymbol{\theta}_m) + \lambda_c \sum_{i=1}^{n} \mathcal{L}_c(\boldsymbol{\theta}_c), \tag{6}$$

where $\boldsymbol{\theta}$ stands for all parameters in MAIC, and λ_m and λ_c are adjustable hyperparameters of the teammate modeling loss and sparse regularization respectively. In the CTDE framework, the mixing network will be removed during the execution phase. To prevent the lazy-agent problem (Sunehag et al. 2018) and reduce model complexity, we make the local network, including teammate modeling and the message generator, have same parameters for all agents.

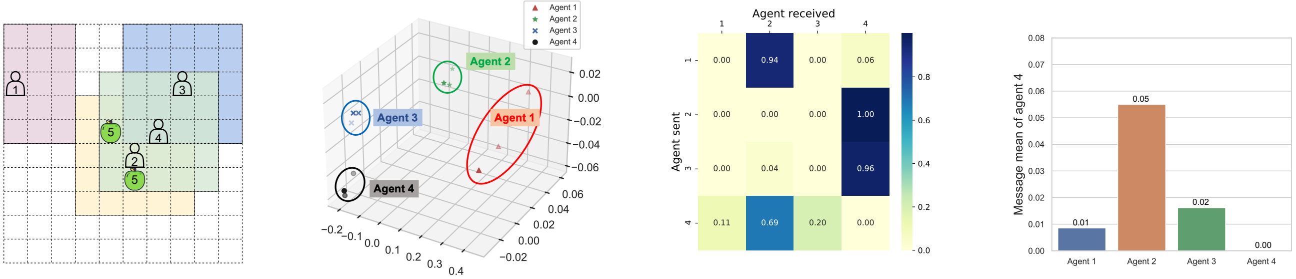
Experimental Results

In this section, we conduct experiments on multiple cooperative MARL environments, aiming to verify whether MAIC can provide more effective and targeted communication.[1] We evaluate MAIC with multiple state-of-the-art baselines on tasks including level-based foraging, Hallway, and the StarCraft II unit micromanagement benchmark. As MAIC can be applied to any value factorization method, we choose the simple VDN mixing network on the first two small cooperative tasks and the more complex QMIX mixing network on the StarCraft II unit micromanagement benchmark. All presented curves are illustrated with average performance and $25 \sim 75\%$ deviation over five random seeds.

Level-Based Foraging

Level-based foraging (LBF) (Papoudakis et al. 2021) is a partially observable grid world game, where agents and foods are initialized with random skill levels. The action space of each agent consists of the movement in four directions, loading food and a "none" action. A group of agents can collect the food if they all choose the loading food action and the summation of their levels is greater than or equal to the level of the food. Then agents will receive a reward correlated to the level of the food. The goal of agents is to maximize the global return in a limited horizon, and the maximized return is normalized to one. Figure 2(a) shows our conducted task, where 4 agents need to cooperate in collecting 2 portions of

[1] Code available at https://github.com/mansicer/MAIC



(a) A test frame on level-based foraging.

(b) Visualization of teammate representation by PCA projection.

(c) Communication weights of different agent pairs.

(d) Average message values of Agent 4 from the test frame.

Figure 2: A case study on the level-based foraging task indicates that MAIC can generate different communication weights with local information to realize targeted communication by teammate modeling. (a) One test frame from the level-based foraging environment. (b) Visualization of every agent's teammate representation from all other agents' teammate modeling based on the test frame in (a). The dimensionality of representations is reduced by PCA. (c) The communication weights generated by each agent to all other agents based on the test frame in (a). The y-axis stands for the agent who generates these communication weights, and the x-axis stands for the agent who will receive the corresponding message. (d) The average message value generated by Agent 4 to any other agent is based on the test frame in (a). The presented scalar is averaged from the original message vector.

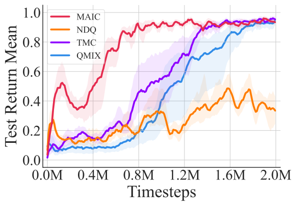
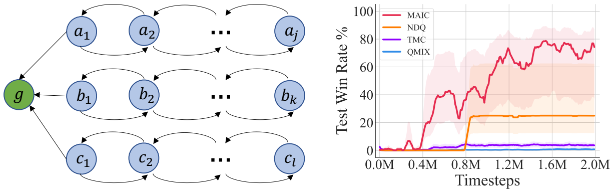


Figure 3: Average returns on level-based foraging.



(a) Task Hallway.

(b) Results on Hallway.

Figure 4: (a) Hallway; (b) Average battle win rate on Hallway.

food in a 10 × 10 grid world with a limited horizon of 50, agents have a restricted vision with range 2. The difference of levels between agents and foods makes strong coordination necessary towards receiving a high return.

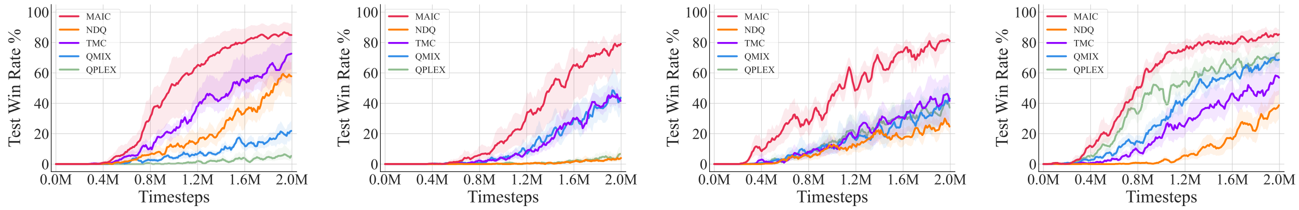
We compare MAIC with multiple baselines, including the classical CTDE method QMIX and two state-of-the-art MARL communication methods named NDQ and TMC. As shown in Figure 3, MAIC gets an averaged return close to one at 0.8M timesteps, faster than all other baselines, which shows the efficient communication structure of MAIC accelerates coordination. TMC presents the advantage of communication compared with QMIX. NDQ struggles in LBF, probably because the augmentation of policy space and the uncertainty of messages lead to insufficient training.

We further investigate how the components of MAIC agents help coordinate. We evaluate MAIC in multiple tests, and Figure 2(a) shows one frame from a test episode, where the fill color indicates a sight range of 2 for each agent. Figure 2(b) correspondingly shows the latent teammate representations computed by each agent individually, where we apply dimensionality reduction to original representations by principal component analysis (PCA) (Wold, Esbensen, and Geladi 1987). We can find that there is much more variance among the teammate representations of Agent 1, as Agent 1 is out of sight for other agents in few timesteps, resulting in ambiguous modeling. The representations of Agents 2, 3, and 4 have less discrepancy, and their internal variances are much smaller because they can see each other more frequently.

Figure 2(c) exhibits values of different communication weight pairs in this test frame. Agents 2 and 4 coordinate strongly since their communication weights are quite large. Agent 4 also has non-zero weights with other agents as it can coordinate with any other agent to successfully get the food with the level of 5. Agent 3 presents a huge communication weight toward Agent 4, for it considers Agent 4 has the most probability to acquire the food. Nevertheless, Agent 1 generates the communication weight in a slightly arbitrary way because it is a little unclear about other agents due to the long distance. The average values of messages from Agent 4 to other agents are shown in Figure 2(d), which illustrates that Agent 4 generates different but targeted messages correlated to its communication weights of other agents.

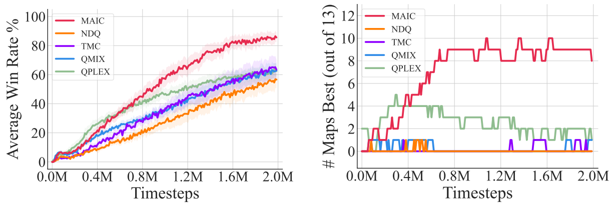
Hallway

Hallway (Wang et al. 2020d) (Figure 4(a)) is a sparse reward cooperative environment with 3 agents randomly initialized at states a_1 to a_j, b_1 to b_k and c_1 to c_l, respectively. An agent can only observe its own position and select actions from moving left, moving right, and keeping still. The episode ends if some agents arrive at state g, and they win and get a reward of 10 only when reaching state g simultaneously. The horizon is set to $\max(j, k, l) + 10$ for avoiding infinite loops.



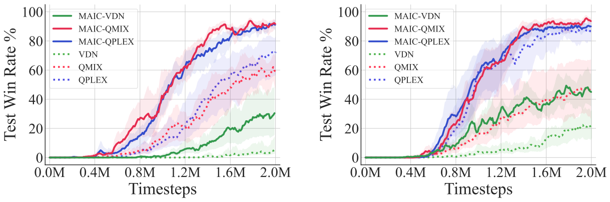
(a) 3c_vs_100zg (b) MMM3 (c) 27m_vs_30m (d) 1c3s5z_vs_1c3s6z

Figure 5: Average test win rates on four SMAC super hard maps.



(a) Average test win rate. (b) Number of best maps.

Figure 6: (a) The average test win rate across all 13 scenarios; (b) the number of scenarios (out of all 13 maps) where the algorithm's average test win rate is the highest by an advantage of at least 0.1% (smoothed in a range of 5 timesteps).



(a) MMM2 (b) 2c_vs_64zg

Figure 7: Average test win rates of MAIC implemented with mixing networks including VDN, QMIX, and QPLEX.

We set $j = 2, k = 6, l = 10$ to make the simultaneous arrival difficult, as agents need strong coordination and communication to win in this partially observable task. As shown in Figure 4(b), QMIX, NDQ, and TMC almost fail to resolve the Hallway problem. QMIX fails because agents cannot acquire other teammates' positions and actions, leading to the failure of policy learning, which indicates efficient communication is necessary for an extremely partially observable environment. TMC also fails on the Hallway task, as the message broadcast paradigm cannot adapt to agents with quite different states and causes miscoordination issues. NDQ may learn a valid policy in few running seeds, but its average performance is much worse than MAIC, which resolves the task with more effective communication.

StarCraft II Micromanagement Benchmark

We applied our method and baselines on the StarCraft II Multi-Agent Challenge (SMAC) benchmark (Samvelyan et al. 2019). We test all methods in 13 maps, including three easy maps, three hard maps, and seven super hard maps. The compared baselines include QMIX, NDQ, TMC, and QPLEX (Wang et al. 2021a), a novel value decomposition method that reaches the state-of-the-art performance in the SMAC benchmark. All hyperparameters for training and in-game AI are the same as PyMARL[2] on StarCraft 2.4.6.

To demonstrate the overall performance of each algorithm, Figure 6 shows the average test win rates across all 13 maps and the number of maps where an algorithm outperforms others, respectively. From Figure 6(a), we can find MAIC achieves outstanding performance compared with other baselines and converges faster than methods except for QPLEX in the initialed 0.5M timesteps. We guess the improvement in the representation capability of QPLEX makes it good at easy and medium maps at the beginning of training. When the training finishes, MAIC exceeds nearly 20% averaged win rate in all 13 scenarios. NDQ has worse overall performance than QMIX, which may be attributed to its augmented policy space and the complex constraints on message learning, resulting in higher training cost. TMC, QPLEX, and QMIX have comparable performance when converged, while TMC and QPLEX's performances are a little better than QMIX for their adaptions on QMIX. Figure 6(b) illustrates MAIC finally has the best performance in up to 10 scenarios among all scenarios and is worse than QPLEX and other baselines in 3 scenarios, while almost tying with others in the rest.

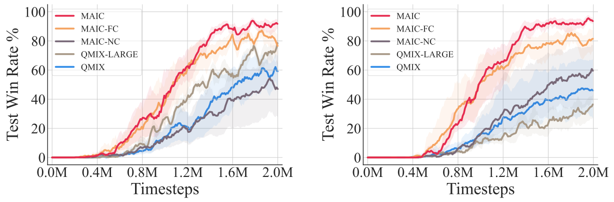
Figure 5 shows the learning curves on four super hard maps in SMAC. MAIC significantly outperforms other baselines on these super hard maps, showing its high coordination ability even in complex settings. Notably, NDQ fails on maps with many agents like MMM3 and 27m_vs_30m as it is difficult to process a large number of messages while MAIC can handle numerous messages in an incentive way. TMC also exhibits less promising performance because its broadcast paradigm limits its ability, but MAIC's tailored communication provides richer representative ability for better coordination.

Integrative Abilities MAIC is agnostic to specific value decomposition methods. We can integrate it with existing MARL value decomposition methods such as VDN, QMIX, and QPLEX. The combined methods, called MAIC-VDN, MAIC-QMIX, and MAIC-QPLEX, respectively, are tested on maps including MMM2 and 2c_vs_64zg. As shown in Fig-

[2] https://github.com/oxwhirl/pymarl

Message	MAIC		TMC		NDQ		VBC	
Pruning	Test Win	Win Rate	Test Win	Win Rate	Test Win	Win Rate	Test Win	Win Rate
Rate	Rate %	Increase	Rate %	Increase	Rate %	Increase	Rate %	Increase
0%	**96.7 ± 0.50**	0.00%	85.6 ± 5.45	0.00%	72.5 ± 6.37	0.00%	59.8 ± 1.05	0.00%
10%	**96.2 ± 0.63**	-0.52%	85.0 ± 5.38	-0.73%	35.0 ± 11.2	-51.7%	60.0 ± 0.74	**0.33%**
20%	**96.8 ± 0.50**	0.12%	86.2 ± 4.24	**0.73%**	27.5 ± 7.23	-62.1%	58.7 ± 1.44	-1.77%
30%	**96.4 ± 0.62**	**-0.31%**	83.1 ± 7.02	-2.92%	27.5 ± 3.64	-62.1%	59.4 ± 2.20	-0.57%
70%	**96.8 ± 0.63**	**0.12%**	81.9 ± 6.06	-4.38%	18.8 ± 10.3	-74.1%	59.2 ± 1.46	-1.27%
80%	**97.1 ± 0.98**	**0.43%**	79.5 ± 5.51	-7.12%	19.4 ± 6.37	-73.3%	58.9 ± 0.90	-1.47%
90%	**96.5 ± 0.55**	**-0.27%**	78.7 ± 5.00	-8.06%	7.50 ± 4.24	-89.7%	57.8 ± 2.23	-3.35%

Table 1: Average test win rates and average win rate changes for MAIC, TMC, NDQ, and VBC under different message pruning rates. The results are averaged from 1000 test episodes among 5 random seeds.



(a) MMM2

(b) 2c_vs_64zg

Figure 8: Average test win rates for ablations studies.

ure 7, we find all integrated methods outperform the original methods, which indicates the communication paradigm of MAIC can significantly enhance coordination.

Ablation Studies To evaluate the effect of every component in MAIC, we design the ablation studies with multiple compared algorithms. We call the MAIC version without sparse communication control MAIC-FC, as it continuously sends tailored messages to other teammates with uniform communication weights. Furthermore, the MAIC version without teammate modeling is called MAIC-NC, whose agent broadcasts non-tailored messages to its teammates due to the lack of teammate modeling. We also introduce QMIX-LARGE, which has a similar number of parameters as MAIC, to investigate whether the superiority of MAIC over QMIX is due to the increase in the number of parameters. We conduct experiments on the same two maps, 2c_vs_64zg and MMM2, respectively. As shown in Figure 8, the comparison between MAIC and QMIX-LARGE indicates that QMIX with a larger network cannot fundamentally improve the performance, and the larger network even leads to a lower win rate in MMM2. MAIC-FC outperforms all methods except MAIC, which shows tailored messages make sense, but the broadcast communication paradigm leads to redundancy and injury to the learning process. MAIC-NC has comparable performance as QMIX, which indicates communication without intentionality does not make sense.

Results on Message Pruning We now investigate the robustness of MAIC by message pruning. We conducted experiments on MMM2, which is a super hard map requiring strong coordination. Communication methods including MAIC, TMC, NDQ, and VBC, a communication method with variance-based control, are trained for 2M timesteps. We test the final models after 2M timesteps to evaluate the test win rates under different message pruning rates. For a fair comparison, we prune the sent messages by its values in an ascending order accordingly with a given pruning rate, except for VBC, which is a particular case as VBC can prune messages with the proposed variance-based control. Table 1 shows the test win rates and performance changes of all algorithms under different message pruning rates. We can find the performance of MAIC is better than other baselines under any message pruning rate. The win rate of NDQ decreases dramatically, as the reduction of messages will severely alter the input of the policy. TMC has higher robustness than NDQ because TMC adds two regularizers to achieve robustness. VBC shows a more robust communication for its variance-based pruning way, but its win rates are not much promising. MAIC has a smaller win rate decrease under most pruning rates and even performs better than the complete communication paradigm, suggesting that MAIC is sufficiently robust to cut useless communication links and the elimination of redundant messages can promote coordination.

Conclusion

This paper proposes MAIC, a novel framework which adopts incentive communication via teammate modeling to enhance coordination. We utilize teammate modeling to anticipate action selection of other agents, guiding the incentive message generation, and then introduce a tailored communication structure by computing communication weights towards different agents. Empirical results in diverse multi-agent cooperative tasks show that our method significantly outperforms other baselines and can be integrated with existing value decomposition methods to improve coordination. Furthermore, we find MAIC can prune most messages without an evident performance decrease, indicating its high robustness. MAIC learns teammate modeling for each agent, which may be intractable when facing environments with hundreds or thousands of agents. Solving the scalability issue by techniques like grouping would be of great interest, and more discussions on efficient communication paradigms are promising.

Acknowledgement

This work is supported by National Key Research and Development Program of China (2020AAA0107200), NSFC (61921006, 61876119), Fundamental Research Funds for the Central Universities (022114380010), and Science and Technology Innovation 2030 – "New Generation Artificial Intelligence" Major Project (No. 2018AAA0100904). The authors would like to thank Rong-jun Qin, Lanjihong Ma, Jia-han Cao, Feng Chen, and the anonymous reviewers for their helpful discussions and support.

References

Albrecht, S. V.; and Stone, P. 2018. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258: 66–95.

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. 2017. Deep variational information bottleneck. In *International Conference on Learning Representations*.

Cao, J.; Yuan, L.; Wang, J.; Zhang, S.; Zhang, C.; Yu, Y.; and Zhan, D.-C. 2021. LINDA: Multi-agent local information decomposition for awareness of teammates. arXiv:2109.12508.

Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734.

Das, A.; Gervet, T.; Romoff, J.; Batra, D.; Parikh, D.; Rabbat, M.; and Pineau, J. 2019. TarMAC: Targeted multi-agent communication. In *International Conference on Machine Learning*, 1538–1546.

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

Du, W.; and Ding, S. 2021. A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications. *Artificial Intelligence Review*, 54(5): 3215–3238.

Du, Y.; Liu, B.; Moens, V.; Liu, Z.; Ren, Z.; Wang, J.; Chen, X.; and Zhang, H. 2021. Learning correlated communication topology in multi-agent reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, 456–464.

Foerster, J. N.; Assael, Y. M.; de Freitas, N.; and Whiteson, S. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2137–2145.

Foerster, J. N.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2974–2982.

Giles, C. L.; and Jim, K.-C. 2002. Learning communication for multi-agent systems. In *Workshop on Radical Agent Concepts*, 377–390. Springer.

Hernandez-Leal, P.; Kartal, B.; and Taylor, M. E. 2019. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 33(6): 750–797.

Kraemer, L.; and Banerjee, B. 2016. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190: 82–94.

Lazaridou, A.; and Baroni, M. 2020. Emergent multi-agent communication in the deep learning era. arXiv:2006.02419.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, 6379–6390.

Lyu, X.; Xiao, Y.; Daley, B.; and Amato, C. 2021. Contrasting centralized and decentralized critics in multi-agent reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, 844–852.

Mao, H.; Zhang, Z.; Xiao, Z.; Gong, Z.; and Ni, Y. 2020a. Learning agent communication under limited bandwidth by message pruning. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, 5142–5149.

Mao, H.; Zhang, Z.; Xiao, Z.; Gong, Z.; and Ni, Y. 2020b. Learning multi-agent communication with double attentional deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 34(1): 1–34.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Nowé, A.; Vrancx, P.; and De Hauwere, Y.-M. 2012. Game Theory and Multi-agent Reinforcement Learning. In *Reinforcement Learning*, 441–470. Springer.

Oliehoek, F. A.; and Amato, C. 2016. *A Concise Introduction to Decentralized POMDPs*. Springer.

Papoudakis, G.; Christianos, F.; Schäfer, L.; and Albrecht, S. V. 2021. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. arXiv:2006.07869.

Rashid, T.; Samvelyan, M.; Schroeder, C.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2018. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, 4295–4304.

Samvelyan, M.; Rashid, T.; de Witt, C. S.; Farquhar, G.; Nardelli, N.; Rudner, T. G. J.; Hung, C.-M.; Torr, P. H. S.; Foerster, J.; and Whiteson, S. 2019. The StarCraft multi-agent challenge. arXiv:1902.04043.

Singh, A.; Jain, T.; and Sukhbaatar, S. 2019. Learning when to communicate at scale in multiagent cooperative and competitive tasks. In *International Conference on Learning Representations*.

Son, K.; Kim, D.; Kang, W. J.; Hostallero, D. E.; and Yi, Y. 2019. QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, 5887–5896.

Stenning, K.; Lascarides, A.; and Calder, J. 2006. *Introduction to Cognition and Communication*. MIT Press.

Sukhbaatar, S.; Szlam, A.; and Fergus, R. 2016. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, 2244–2252.

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V. F.; 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 *International conference on Autonomous Agents and Multiagent Systems*, 2085–2087.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. MIT Press.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Wang, J.; Ren, Z.; Liu, T.; Yu, Y.; and Zhang, C. 2021a. QPLEX: Duplex dueling multi-agent Q-learning. In *International Conference on Learning Representations*.

Wang, J.; Zhang, Y.; Kim, T.-K.; and Gu, Y. 2020a. Shapley Q-value: A local reward approach to solve global reward games. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, 7285–7292.

Wang, T.; Dong, H.; Lesser, V.; and Zhang, C. 2020b. ROMA: Multi-agent reinforcement learning with emergent roles. In *International Conference on Machine Learning*, 9876–9886.

Wang, T.; Gupta, T.; Mahajan, A.; Peng, B.; Whiteson, S.; and Zhang, C. 2020c. RODE: Learning roles to decompose multi-agent tasks. In *International Conference on Learning Representations*.

Wang, T.; Wang, J.; Zheng, C.; and Zhang, C. 2020d. Learning nearly decomposable value functions via communication minimization. In *International Conference on Learning Representations*.

Wang, Y.; Han, B.; Wang, T.; Dong, H.; and Zhang, C. 2021b. DOP: Off-policy multi-agent decomposed policy gradients. In *International Conference on Learning Representations*.

Wold, S.; Esbensen, K.; and Geladi, P. 1987. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1-3): 37–52.

Xue, W.; Qiu, W.; An, B.; Rabinovich, Z.; Obraztsova, S.; and Yeo, C. K. 2021. Mis-spoke or mis-lead: Achieving robustness in multi-agent communicative reinforcement learning. arXiv:2108.03803.

Zhang, C.; and Lesser, V. 2011. Coordinated multi-agent reinforcement learning in networked distributed POMDPs. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 764–770.

Zhang, S. Q.; Zhang, Q.; and Lin, J. 2019. Efficient communication in multi-agent reinforcement learning via variance based control. In *Advances in Neural Information Processing Systems*, 3230–3239.

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

Zhang, T.; Li, Y.; Wang, C.; Xie, G.; and Lu, Z. 2021. FOP: Factorizing optimal joint policy of maximum-entropy multi-agent reinforcement learning. In *International Conference on Machine Learning*, 12491–12500.

Zhou, M.; Luo, J.; Villela, J.; Yang, Y.; Rusu, D.; Miao, J.; Zhang, W.; Alban, M.; Fadakar, I.; Chen, Z.; et al. 2020. SMARTS: Scalable multi-agent reinforcement learning training school for autonomous driving. In *Conference on Robot Learning*.