SCIENCE CHINA
Information Sciences

• **RESEARCH PAPER** •

Robust Cooperative Multi-agent Reinforcement Learning via Multi-view Message Certification

Lei YUAN[1, 2], Tao JIANG[1], Lihe LI[1], Feng CHEN[1], Zongzhang ZHANG[1] & Yang YU[1, 2*]

[1]*National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210000, *China;*
[2]*Polixir Technologies, Nanjing* 211106, *China*

Abstract Many multi-agent scenarios require message sharing among agents to promote coordination, hastening the robustness of multi-agent communication when policies are deployed in a message perturbation environment. Major relevant works tackle this issue under specific assumptions, like a limited number of message channels would sustain perturbations, limiting the efficiency in complex scenarios. In this paper, we take a further step in addressing this issue by learning a **ro**bust **C**ooperative **M**ulti-**A**gent Reinforcement Learning via Multi-view Message **C**ertification, dubbed **CroMAC**. Agents trained under CroMAC can obtain guaranteed lower bounds on state-action values to identify and choose the optimal action under a worst-case deviation when the received messages are perturbed. Concretely, we first model multi-agent communication as a multi-view problem, where every message stands for a view of the state. Then we extract a certificated joint message representation by a multi-view variational autoencoder (MVAE) that uses a product-of-experts inference network. For the optimization phase, we do perturbations in the latent space of the state for a certificate guarantee. Then the learned joint message representation is used to approximate the certificated state representation during training. Extensive experiments in several cooperative multi-agent benchmarks validate the effectiveness of the proposed CroMAC.

Keywords Multi-agent Reinforcement Learning, Robust Communication, Adversarial Training, Multi-view Learning, Message Certification

Citation Lei YUAN, Tao JIANG, Lihe LI, Feng CHEN, Zongzhang ZHANG, Yang YU. Robust Cooperative Multi-agent Reinforcement Learning via Multi-view Message Certification. Sci China Inf Sci, for review

1 Introduction

Many real-world problems are made up of multiple interactive agents, which could usually be modeled as a Multi-Agent Reinforcement Learning (MARL) problem [3, 89]. Further, when the agents hold a shared goal, this problem refers to cooperative MARL [42], which shows great progress in diverse domains like power management [65], multi-UAV control [86], dynamic algorithm configuration [78], etc. Many methods are proposed to promote the coordination ability of MARL, including value-based methods [50, 57, 64], policy-gradient-based methods [32, 81, 82], and some variants [4, 67, 69, 84], showing great progress in many complex and challenging benchmarks [17, 45]. Nevertheless, prior works hugely depend on the strength of Deep Neural Networks (DNNs), whose vulnerability might cause catastrophic results when any perturbation happens [40]. Recently this phenomenon has been tested in cooperative MARL [20], showing that a cooperative MARL system is of low robustness when encountering any perturbations (e.g., state, action, and reward).

* Corresponding author (email: yuy@nju.edu.cn)

Robustness has been widely investigated in single-agent reinforcement learning (RL) [40], and many works have applied different techniques to various aspects to investigate it. A prior popular way is to introduce an auxiliary adversary to play against the ego-system [47, 52, 62, 87], then model the process of policy learning as a minimax problem from the perspective of game theory [83], which may trigger performance deterioration or even unsafe behaviors when facing an unpredictable adversarial policy. Another kind of method tackles this issue by designing efficient and useful regularizers in the training process [41, 56, 72, 87], showing efficient robustness in various domains. Certificate-based methods furthermore apply some techniques like vector-ϵ-ball perturbations to obtain a certificate robustness guarantee during the training and testing phases [12, 54, 70]. However, the MARL problem differs considerably from the single-agent setting, with multiple agents interacting with others [11].

For the robustness of MARL, new challenges such as scalability [8] arise as multiple agents interact with others in the training phase. For example, in the auxiliary adversary training paradigm, the action space of adversary policy may grow dramatically with respect to the number of agents in an MARL system. Works on robust MARL should then consider both the robustness and the multi-agent specificity. Some works design efficient mechanisms to obtain a robust policy to avoid overfitting to specific partners [61] or opponents [28], and others consider the Markov decision process (MDP) itself to get a robust policy in response to state [92], reward [88], and action [21, 22]. Nonetheless, the robustness of a communication policy is much more complex [93], as we should consider **when** to give **what** perturbations on **which** message channel(s) to adversarially train the communication policy. Prior works mainly investigate the emergence of adversarial communication [2] or impose constraints like a limited number of message channels [55, 79] suffering from message perturbations. These approaches make progress somewhat, but the constraints hinder the robustness's completeness and are also far away from the real-world condition, as all the message channels could sustain perturbations [35]. Even worse, these approaches lack formal robustness guarantees or certificates between each agent's received messages and decision-making.

With this in mind, we propose to promote robustness in current multi-agent communication methods. We posit obtaining a robust communication policy where **every messaging channel** could suffer from **perturbations** at **any time**. Specifically, for any agent in an N-agent system, it will receive $N-1$ messages. As each message is a different view of the state, we model the message-receiving process as a multi-view (a.k.a. "multi-modal") problem, then obtain joint message representations with robustness guarantees from each received message by a Multi-view Variational AutoEncoder (MVAE) that uses a product-of-experts inference network. For the optimization phase, we first encode the state into a latent space and do perturbations in this space to obtain a certificate relationship between the latent variable and the agents' Q-values. Then, we train the message representation by approximating the certificated latent variables, and ensure certification between each message and the agents' Q-value implicitly. As we directly impose perturbations in the latent space, the problem of specific action designing for any auxiliary adversaries can be avoided. For evaluation, we conduct extensive experiments on various cooperative multi-agent benchmarks, including Hallway [66], Level-Based Foraging [45], Traffic Junction [9], and two StarCraft Multi-Agent Challenge (SMAC) maps [66]. The results show that CroMAC achieves comparable or superior performance to multiple baselines. Moreover, visualization results show how CroMAC works, and more results demonstrate its high generality ability for different methods under different conditions.

2 Related Work

Multi-Agent Communication plays a promising role in multi-agent coordination under partial observability, which considers **when** to communicate with **whom** and **what** contents to share [93]. The early relevant works mainly consider designing different communication paradigms to improve communication efficiency [14, 53]. DIAL [14] is a simple communication mechanism where agents broadcast messages to all teammates, allowing the gradient to flow among agents for end-to-end training with reinforcement learning. CommNet [53] 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.

As the mentioned communication paradigm may cause message redundancy, some works employ techniques such as gate mechanisms [10, 37, 77] to explicitly decide whom to communicate with, or attention mechanisms [9, 38, 68] to weigh different messages. What messages to share among agents is another crucial issue. The most naive way is only to share local observations or their embeddings [14, 66], which inevitably causes bandwidth wasting or even degrades coordination efficiency. Towards a more efficient communication protocol, some methods utilize techniques like teammate modeling to generate more succinct and efficient messages [85, 90, 91]. For the robustness of message sharing in CMARL, Blumenkamp and Prorok [2] develop a new multi-agent learning model that integrates heterogeneous, potentially self-interested policies that share a differentiable communication channel to elicit the emergence of adversarial communications. Xue et al. [79] consider multi-agent adversarial communication, learning robust communication policy when some message senders are poisoned. A recent method named AME [55] is proposed to acquire a robust communication policy when less than half of the agents in the system sustain noise and potential attackers.

Robustness in Single Agent Reinforcement Learning Moos et al. [40] involve perturbations that occur of different aspects in single agent reinforcement learning such as state, reward, policy, etc. Some prior methods introduce an adversary to achieve robustness via training the ego-system and the adversary in an alternative way [43, 47, 52, 62, 87]. RARL [47] picks out specific robot joints that the adversary acts on to find an equilibrium of the minimax objective using an alternative learning adversary. RAP [62] and GC [52] improve RARL by learning population-based augmentation to the Robust RL formulation. However, while these approaches provide better robust policies, it has been shown that such approaches can negatively impact policy performance in non-adversarial scenarios. Moreover, many unsafe behaviors may be exhibited during online attacks, potentially damaging the system controlled by the learning agent if adversarial training occurs in a physical rather than a simulated environment. Other methods improve robustness by designing useful and appropriate regularizers in the loss function [30, 41, 56]. Zhang et al. [87] formulate the problem of decision making under adversarial attacks on state observations as SA-MDP and learn a state-adversarial policy for multiple DRL methods like DDPG and DQN. RADIAL-RL [41] trains reinforcement learning agents with improved robustness against l_p-norm bounded adversarial attacks, showing superior performance on multiple benchmarks. The mentioned approaches achieve robustness compared to adversarial training, improving the sample efficiency as they need not train an auxiliary adversary. Furthermore, these mentioned methods lack theoretical guarantee, hastening some recent certificate robustness methods [12, 49, 70, 71]. CARRL [49] develops an online certifiably robust policy that computes guaranteed lower bounds on state-action values during execution to identify and choose a robust action under a worst-case deviation in input space due to possible adversaries or noise. CROP [70] gives a solid theoretical guarantee for robust reinforcement learning and applies function smoothing techniques to train a robust policy.

Multi-View (Modal) Representation Learning aims to learn feature representations from multi-view data using different views' information. Its main difficulty is to explicitly measure the content similarity between the heterogeneous samples. How to solve this problem roughly divides multi-view representation learning into three methods: alignment representation [46], joint representation [6], as well as shared and specific representation [75]. The key ideas of these methods are the same, which is establishing a common representation space by exploring the semantic relationship among the multi-view data. One popular and promising way is to use generative models like VAE [24], which generate this representation space in two ways: cross-view generation and joint-view generation. The former learns a conditional generative model over all views by applying techniques like conditional VAE [51]. Nevertheless, the latter learns the joint distribution of the multi-view data. For example, MVAE [73] models the joint posterior as a product-of-experts (POE), and JMVAE [58] learns a shared representation with a joint encoder. Please refer to [1, 80] for a comprehensive review. After the representation space is established by multi-view learning, some approaches use it to solve the modality missing problem [34], or obtain a compact representation from incomplete views [74]. Li et al. [27] extend the partially observable

Markov decision processes (POMDPs) to support more than one observation model and propose two solutions through observation augmentation and cross-view policy transfer in a reinforcement learning problem. DRIBO [13] leverages the sequential nature of RL to learn robust representations that encode only task-relevant information from observations based on the unsupervised multi-view setting. Kinose et al. [25] introduce a novel reinforcement learning agent for integrated recognition and control from multi-view observations. To the best of our knowledge, none of any MARL approaches use multi-view learning to train the communication policy. We take a further step in this direction to get a robust message representation.

Multi-agent Robustness Robustness also plays a promising role in MARL [20], but suffers from extra challenges that do not appear in the single-agent setting, as interactions exist among agents [19], leading to new and specific considerations such as non-stationarity [44], credit assignment [63], and scalability [8] when improving the robustness of any multi-agent system [20]. One type of relevant work aims to investigate the robustness of a learned coordination policy. Lin et al. [31] first learn an observation attacker via RL, then use it to poison one manually selected agent, showing the multi-agent system is vulnerable to observation perturbation. Guo et al. [20] recently do more comprehensive robustness testing on reward, state, and action for typical MARL methods like QMIX [50] and MAPPO [82]. As for robustness improvement in MARL, research is conducted on multiple aspects. Many prior works focus on designing an efficient approach to learning a robust coordination policy to avoid overfitting to specific partners [61] or opponents [28]. Akin to considering the MDP in a single-agent setting (e.g., state, reward, action), R-MADDPG [88] considers the model uncertainty of an MARL system, then introduces the concept of robust Nash equilibrium. Hu et al. [21] apply a heuristic rule to investigate the robustness of MARL when some agents suffer from action mistakes, and utilize correlated equilibrium theory to learn a robust coordination policy. Robustness in multi-agent communication has also attracted some attention in recent years. Mitchell et al. [39] apply a filter based on the Gaussian process to extract valuable content from noisy messages. Tu et al. [60] study robustness at the neural network level for secure multi-agent systems. Xue et al. [79] model multi-agent communication as a two-player zero-sum game and apply the policy-search response-oracle (PSRO) technique to learn a robust communication policy. The most related work to ours is Ablated Message Ensemble (AME) [55], which assumes no more than half of the message channels in the system may be attacked, then introduces an ensemble-based defense method to achieve robustness. However, we will show that this approach performs poorly in complex scenarios, as the constraints may impede robust efficiency.

3 Problem Formulation

We consider a fully cooperative MARL communication problem, which can be formally modeled as a Decentralized Partially Observable MDP under Communication (Dec-POMDP-Com) [79] and formulated as a tuple $\langle \mathcal{N}, \mathcal{S}, \mathcal{A}, P, \Omega, O, R, \gamma, \mathcal{M} \rangle$, where $\mathcal{N} = \{1, \ldots, n\}$, $\mathcal{S}$, $\mathcal{A}$, and Ω are the sets of agents, states, actions, and observations, respectively. O is the observation function, P denotes the transition function, R represents the reward function, $\gamma \in [0, 1)$ stands for the discounted factor, and $\mathcal{M}$ indicates the message set. Due to the partially observable nature of the environment, each agent $i \in \mathcal{N}$ can only obtain the local observation $o_i \in \Omega$, and hold an individual policy $\pi(a_i \mid \tau_i, m_i)$, where τ_i represents the output of a trajectory encoder (e.g., GRU [7]) which encodes $(o_i^1, a_i^1, \ldots, o_i^{t-1}, a_i^{t-1}, o_i^t)$, and $m_i \in \mathcal{M}$ is the messages received by agent i and m_{ij} represents the message transmitted from j to i. As each agent can behave as a message sender as well as a message receiver, this paper considers learning useful message representation on the receiving end, and agents only use local information (e.g., τ_i, and we use $m_{:,i}$ for generality) as message $m_{:,i}$ to share within the team. We aim to find an optimal policy under the setting where each message channel in the multi-agent system may suffer from perturbations. In line with the widely used state-adversarial MDP (SA-MDP) in single-agent RL [48, 87], we formulate this setting as a message-adversarial Dec-POMDP-Com (MA-Dec-POMDP-Com).

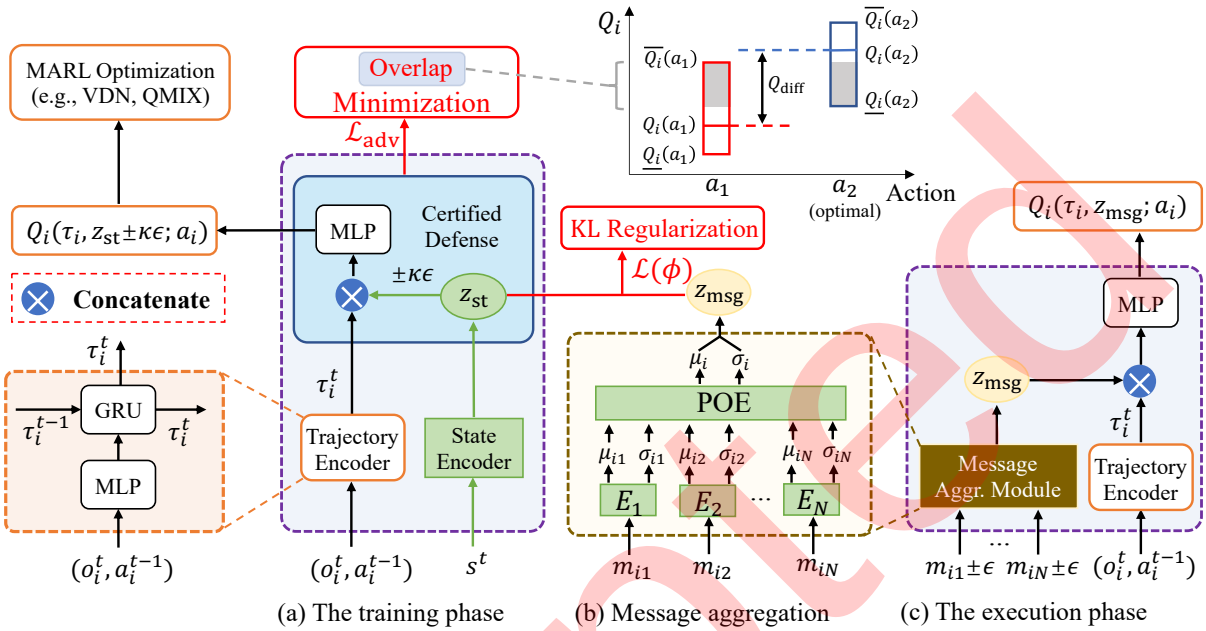


Figure 1 Structure of CroMAC. (a) During the training phase, we encode the state into latent variables $z_{\rm st}$, then perturb it to gain a certificate guarantee between $z_{\rm st}$ and $Q_i(\tau_i, z_{\rm st} \pm \kappa\epsilon; a_i)$, and this process is optimized via minimizing the overlap (i.e., the gray part) weighted by the difference of Q-values. Here a_2 represents the original optimal action while a_1 is another sub-optimal action. $\overline{Q}$ and $\underline{Q}$ represent the upper bound and lower bound of Q-value under perturbation respectively, therefore the gray overlap measures the probability of selecting action besides the original optimal action under perturbation. Since not all overlap is equally important, if two actions have similar Q-values, i.e., $Q_{\rm diff}$ is small, we can ignore the overlap as taking a different but equally good action under perturbation is acceptable. The whole process can be optimized by any value decomposition methods like QMIX [50], and the output of the message aggregation module $z_{\rm msg}$ is then used to approximate $z_{\rm st}$ by minimizing their distance (e.g., KL divergence). (b) The message aggregation module. Each message m_{ij} is encoded into a latent space via a message encoder E_j, where $j \in \{1, \cdots, i-1, i+1, \cdots, N\}$, and the parameters of E_j are regularized to obtain certificates between the joint message representation and each message. (c) After training, we use the learned message aggregation module and other shared modules like the trajectory encoder to make a decision in a decentralized way.

In an MA-Dec-POMDP-Com, we introduce a message adversary $v(m) : m \to \hat{m}$. The adversary perturbs the messages received by each agent, such that agent i takes action by $\pi(a_i|\tau_i, \hat{m}_i)$. The joint action $\boldsymbol{a} = \langle a_1, \ldots, a_n \rangle$ leads to the next state $s' \sim P(\cdot \mid s, \boldsymbol{a})$ 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 $Q^{\boldsymbol{\pi}}_{\rm tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \mathbb{E}_{s,\boldsymbol{a}}\left[\sum_{t=0}^{\infty} \gamma^t R(s, \boldsymbol{a}) \mid s_0 = s, \boldsymbol{a_0} = \boldsymbol{a}, \boldsymbol{\pi}\right]$, with $\boldsymbol{\tau} = \langle \tau_1, \ldots, \tau_n \rangle$. If the adversary can perturb a message m arbitrarily without bounds, the problem becomes trivial [76]. To fit our method to the most realistic settings, we restrict the power of an adversary to a perturbation set $\mathcal{B}$, i.e., $\mathcal{B} = \{\hat{m} \mid \|m - \hat{m}\|_p \leqslant \epsilon\}$, where ϵ is the given perturbation magnitude and p determines the type of norm. Our experiments in this paper focus on $p = \infty$. Furthermore, since $\mathcal{B}(m)$ is usually a small set nearby m, our adversary applies FGSM [16] to learn a perturbation vector Δ, and we project $m + \Delta$ to $\mathcal{B}(m)$.

4 Method

This section gives a detailed description of our proposed CroMAC (Fig. 1), a novel approach that achieves a robust communication policy under MA-Dec-POMDP-Com. As each message is a view of the state, we first apply a multi-view variational autoencoder (MVAE) that uses a product-of-experts inference network to extract a joint message representation. Then we obtain the bounds between the joint message representation and each message by interval bound propagation [18]. In the training phase, we first encode state $s_t \in \mathcal{S}$ into a latent variable $z_{\rm st}$, then we impose perturbations in the latent space to gain a certificate guarantee between $z_{\rm st}$ and each state-action value $Q_i(\tau_i, z_{\rm st} \pm \kappa\epsilon; a_i)$, where $\pm\kappa\epsilon$ represents

that the variable suffers from ℓ_∞-norm perturbations within budget $\kappa\epsilon$ and κ is a constant. Finally, the joint message representation z_{msg} is optimized by approximating z_{st} via minimizing the Kullback-Leibler divergence between these two variables, endowing certification between each message and each state-action value implicitly. In the execution phase, we only use the message aggregation module and the trajectory encoder to make decisions in a decentralized way.

4.1 Multi-view Multi-agent Communication

We consider learning a robust communication policy in an MA-Dec-POMDP-Com. For each agent i, there are $N-1$ message channels, thus each agent receives multiple available messages about the environment. Inspired by the widely used multi-view learning [23, 29], we apply the product-of-experts (POE) [18] technique to extract joint message representations. Formally, agent i receives multiple messages m_{ij}^t from teammate $j \in \{1, \cdots, i-1, i+1, \cdots, N\}$ and we denote its local history τ_i^t as m_{ii}^t at time t. We assume each message is conditioned on an unknown hidden variable z_{ij}^t, then the generation of multiple messages can be modeled as a multi-view variational autoencoder process. We then optimize the Evidence Lower BOund (ELBO) to maximize the marginal likelihood with a message encoder $q_{\boldsymbol{\phi}_{\text{enc}}}(z_{ij}^t|m_{ij}^t)$ parameterized with $\boldsymbol{\phi}_{\text{enc}}$, and a message decoder $p_{\boldsymbol{\phi}_{\text{dec}}}(m_{ij}^t|z_{ij}^t)$ with parameter $\boldsymbol{\phi}_{\text{dec}}$:

$$\text{ELBO}(m_{ij}^t) \triangleq \mathbb{E}_{q_{\boldsymbol{\phi}_{\text{enc}}}(z_{ij}^t|m_{ij}^t)}\left[\log p_{\boldsymbol{\phi}_{\text{dec}}}\left(m_{ij}^t \mid z_{ij}^t\right)\right] - \text{KL}\left[q_{\boldsymbol{\phi}_{\text{enc}}}(z_{ij}^t \mid m_{ij}^t), p(z_{ij}^t)\right], \tag{1}$$

where $\text{KL}[p, q]$ is the Kullback-Leibler divergence between distributions p and q. The first term in Eqn. 1 is the reconstruction likelihood, and the second term aims to guarantee that the output of the encoder is similar to the prior distribution $p(z_{ij}^t)$, and can be regarded as a regularization term. The message encoder $q_{\boldsymbol{\phi}_{\text{enc}}}$ outputs parameters of an n-multivariate Gaussian distribution $\mathcal{N}(\mu_{ij}^t, \sigma_{ij}^t)$, where μ_{ij}^t and σ_{ij}^t are the mean and standard deviation of $p(z_{ij}^t)$, respectively. As all messages $\{m_{i1}^t, \cdots, m_{iN}^t\}$ are conditionally independent given the common latent variable z_i^t, we assume a generative model for all the messages in the form:

$$p(m_{i1}^t, \cdots, m_{iN}^t, z_i^t) = p(z_i^t)p(m_{i1}^t|z_i^t)p(m_{i2}^t|z_i^t)\cdots p(m_{iN}^t|z_i^t). \tag{2}$$

Then Eqn. 1 can be extended as:

$$\text{ELBO}(m_i^t) \triangleq \mathbb{E}_{q_{\boldsymbol{\phi}_{\text{enc}}}(z_i^t|m_i^t)}\left[\sum_{j=1}^{N} \log p_{\boldsymbol{\phi}_{\text{dec}}}\left(m_{ij}^t \mid z_i^t\right)\right] - \text{KL}\left[q_{\boldsymbol{\phi}_{\text{enc}}}(z_i^t \mid m_i^t), p(z_i^t)\right], \tag{3}$$

where $m_i^t = \{m_{i1}^t, \cdots, m_{iN}^t\}$ is the set of messages agent i receives at time t and its local history m_{ii}^t (i.e., τ_i^t). Then, this message generation process can be treated as a multi-view representation learning problem [58]. We use the inference network $q(z_i^t \mid m_i^t)$ as a variational distribution to approximate the true posterior $p(z_i^t \mid m_i^t)$, then get the relationship among the joint- and single- view posteriors as:

$$\begin{aligned}
p(z_i^t \mid m_i^t) &= \frac{p(m_i^t \mid z_i^t)p(z_i^t)}{p(m_i^t)} = \frac{p(z_i^t)}{p(m_i^t)}\prod_{j=1}^{N} p(m_{ij}^t \mid z_i^t) \\
&= \frac{p(z_i^t)}{p(m_i^t)}\prod_{j=1}^{N}\frac{p(z_i^t \mid m_{ij}^t)p(m_{ij}^t)}{p(z_i^t)} \\
&= \frac{\prod_{j=1}^{N} p(m_{ij}^t)}{p(m_i^t)}\frac{\prod_{j=1}^{N} p(z_i^t \mid m_{ij}^t)}{\prod_{j=1}^{N-1} p(z_i^t)} \\
&\propto \frac{\prod_{j=1}^{N} p(z_i^t \mid m_{ij}^t)}{\prod_{j=1}^{N-1} p(z_i^t)} \approx \frac{\prod_{j=1}^{N}[q(z_i^t \mid m_{ij}^t)p(z_i^t)]}{\prod_{j=1}^{N-1} p(z_i^t)} \\
&= p(z_i^t)\prod_{j=1}^{N} q(z_i^t \mid m_{ij}^t).
\end{aligned} \tag{4}$$

The last two lines in Eqn. 4 hold as we use $q(z_i^t \mid m_{ij}^t)p(z_i^t)$ to approximate $p(z_i^t \mid m_{ij}^t)$ so that the inference network is composed of N neural networks $q(z_i^t \mid m_{ij}^t)$, and if each view is homogeneous, we can even replace them with only one network with shared parameters. Akin to the standard VAE [24], we apply a deep neural network (e.g., MLP) to model the message encoder $q_{\boldsymbol{\phi}_{\text{enc}}}(z_i^t|m_{ij}^t)$, which outputs the parameters of the Gaussian distribution μ_{ij}, σ_{ij}^2. As for now, we can combine the multiple outputs of message encoder in a simple analytical way: a product of Gaussian experts is itself Gaussian [5]:

$$\begin{aligned} \mu_i &= (\sum_{j=1}^{N} \mu_{ij}\mathrm{T}_{ij})(\sum_{j=1}^{N} \mathrm{T}_{ij})^{-1}, \\ \sigma_i^2 &= (\sum_{j=1}^{N} \mathrm{T}_{ij})^{-1}, \end{aligned} \tag{5}$$

where μ_i and σ_i^2 are the mean and variance of the learned joint message representation's Gaussian distribution, and μ_{ij} and σ_{ij}^2 are the mean and variance of the i-th agent's j-th Gaussian distribution through message encoder, $\mathrm{T}_{ij} = (\sigma_{ij}^2)^{-1}$ is the inverse of the variance. The detailed derivative process can be seen in Appendix A.

4.2 Message Certificates via Bound Propagation

Though we have combined all the received messages into a joint message representation, the learned joint message representation still lacks a certificated guarantee with each received message under perturbation. In this part, we aim to achieve this using the interval bound propagation technique. Formally, consider agent i receives messages $m_i = \{m_{i1}, \cdots, m_{iN}\}$ under perturbation of ℓ_∞-norm attack within given budget ϵ, the upper and lower bounds are $\overline{m_i} = \{\overline{m_{i1}}, \cdots, \overline{m_{iN}}\} = \{m_{i1} + \boldsymbol{\epsilon}, \cdots, m_{iN} + \boldsymbol{\epsilon}\}$ and $\underline{m_i} = \{\underline{m_{i1}}, \cdots, \underline{m_{iN}}\} = \{m_{i1} - \boldsymbol{\epsilon}, \cdots, m_{iN} - \boldsymbol{\epsilon}\}$, respectively. The averages and residuals of the upper and lower bounds can be denoted as:

$$\begin{aligned} \hat{\mu}_0 &= \frac{1}{2}(\overline{m_i} + \underline{m_i}) = m_i, \\ \hat{r}_0 &= \frac{1}{2}(\overline{m_i} - \underline{m_i}) = \boldsymbol{\epsilon}. \end{aligned} \tag{6}$$

Here, we use "average" instead of "mean" to distinguish it from the one in VAE, and $\hat{\mu}$ and $\hat{r}$ are used to represent averages and residuals, respectively. For simplification of notation and mathematical derivation, we assume each message encoder has only one layer fully-connected network with shared parameters, and we can use any NNs with arbitrary depths and element-wise monotonic activation functions (e.g., ReLU, Sigmoid, Tanh) by getting a reasonable bound propagation mechanism [18]. Further, we here use W_m, b_m, and W_v, b_v to represent the parameters of the two separate fully connected layers in the message encoder, which output the means and variances of the Gaussian distributions, respectively. Thus we can propagate the bounds through the one MLP layer by matrix multiplication. The averages and residuals of the bounds come to be:

$$\begin{aligned} \hat{\mu}_m &= \{W_m\hat{\mu}_0(1) + b_m, \cdots, W_m\hat{\mu}_0(N) + b_m\}, \\ \hat{r}_m &= \{|W_m|\hat{r}_0(1), \cdots, |W_m|\hat{r}_0(N)\}, \\ \hat{\mu}_v &= \{W_v\hat{\mu}_0(1) + b_v, \cdots, W_v\hat{\mu}_0(N) + b_v\}, \\ \hat{r}_v &= \{|W_v|\hat{r}_0(1), \cdots, |W_v|\hat{r}_0(N)\}, \end{aligned} \tag{7}$$

where $(\hat{\mu}_m, \hat{r}_m)$ and $(\hat{\mu}_v, \hat{r}_v)$ stand for the (average, residual) pairs of the mean outputs' bounds and the variance outputs' bounds, respectively, and $|\cdot|$ is the element-wise absolute value operator. We use ReLU as the activation function which is element-wise monotonic so we can omit it as it will not affect the propagating bounds. Thus the upper and lower bounds of each single message representation (i.e., mean

and variance) can be written as:

$$\begin{aligned}
\overline{z_m} &= \hat{\mu}_m + \hat{r}_m = \{W_m m_{i1} + b_m + |W_m|\boldsymbol{\epsilon}, \cdots, W_m m_{iN} + b_m + |W_m|\boldsymbol{\epsilon}\},\\
\underline{z_m} &= \hat{\mu}_m - \hat{r}_m = \{W_m m_{i1} + b_m - |W_m|\boldsymbol{\epsilon}, \cdots, W_m m_{iN} + b_m - |W_m|\boldsymbol{\epsilon}\},\\
\overline{z_v} &= \hat{\mu}_v + \hat{r}_v = \{W_v m_{i1} + b_v + |W_v|\boldsymbol{\epsilon}, \cdots, W_v m_{iN} + b_v + |W_v|\boldsymbol{\epsilon}\},\\
\underline{z_v} &= \hat{\mu}_v - \hat{r}_v = \{W_v m_{i1} + b_v - |W_v|\boldsymbol{\epsilon}, \cdots, W_v m_{iN} + b_v - |W_v|\boldsymbol{\epsilon}\}.
\end{aligned} \tag{8}$$

consider the relationship in Eqn. 5, we then have:

$$\begin{aligned}
Z_M &= \big(\sum_i z_m(i) z_v(i)^{-1}\big)\big(\sum_i z_v(i)^{-1}\big)^{-1},\\
Z_V &= \big(\sum_i z_v(i)^{-1}\big)^{-1},
\end{aligned} \tag{9}$$

where Z_M, Z_V represent the mean and variance of the joint message representation. However, we cannot get the upper and lower bounds as the POE we use is not affine neural layers. Notice that Z_V is actually the Harmonic Mean [15] of $\frac{z_v(i)}{N}$ (the Harmonic Mean of variables x_i is $H_n = \frac{n}{\frac{1}{x_1}+\frac{1}{x_2}+\cdots+\frac{1}{x_n}}$) while Z_M is the Weighted Harmonic Mean of $z_m(i)$ with weights $\frac{z_m(i)}{z_v(i)}$ (the Weighted Harmonic Mean of x_i with weights w_i is $H_n = \frac{m_1+m_2+\cdots+m_n}{\frac{m_1}{x_1}+\frac{m_2}{x_2}+\cdots+\frac{m_n}{x_n}}$). Here, variances $z_v(i)$ are non-negative and means $z_m(i)$ can be normalized to a positive range, therefore we can scale Eqn. 9 appropriately by the properties of Harmonic Mean to infer the upper and lower bounds. We here prove one of them for simplicity. Take the variance term for example and simplify W_v, b_v, Z_V, z_v as W, b, Z, z, we have:

$$\begin{aligned}
\overline{Z} &= \big(\sum_i \overline{z}(i)^{-1}\big)^{-1} \leqslant \frac{\mathbf{1}\cdot\max(\overline{z})}{N},\\
\underline{Z} &= \big(\sum_i \underline{z}(i)^{-1}\big)^{-1} \geqslant \frac{\mathbf{1}\cdot\min(\underline{z})}{N}.
\end{aligned} \tag{10}$$

Note that $\mathbf{1}$ is an all-1-vector with the same dimension as $\overline{Z}$. $(\leqslant, \geqslant)$ signs here act on each element of vectors, and $(\max, \min)$ operations find the maximum or minimum number of vectors of the set. We can get the upper bound of the final integration error:

$$\max(\overline{Z} - Z_{\text{true}}, Z_{\text{true}} - \underline{Z}) \leqslant \overline{Z} - \underline{Z} \leqslant \frac{\mathbf{1}\cdot(\max(\overline{z}) - \min(\underline{z}))}{N}, \tag{11}$$

where Z_{true} stands for the ground truth value. Assume the p-th element of the j-th vector of $\overline{z}$ is $\max(\overline{z})$ and q-th element of the k-th vector of $\underline{z}$ is $\min(\underline{z})$, we get

$$\begin{aligned}
\max(\overline{z}) - \min(\underline{z}) &= (Wm_{ij} + b + |W|\boldsymbol{\epsilon})_p - (Wm_{ik} + b - |W|\boldsymbol{\epsilon})_q\\
&= W_{p,:}m_{ij} - W_{q,:}m_{ik} + b_p - b_q + (|W|_{p,:} - |W|_{q,:})\boldsymbol{\epsilon},
\end{aligned} \tag{12}$$

where $W_{p,:}$ means the p-th row of matrix W. We can notice that the integration error can be limited to a constant $\||W|_{p,:} - |W|_{q,:}\|_1/N$ times $\boldsymbol{\epsilon}$ if W, b, m_i are bounded. Through subsequent experiments, we found that good robustness could be achieved when κ times the noise is considered in the integrated information with only W bounded to $[C\_MIN, C\_MAX]$, here C_MIN, C_MAX, κ are hyperparameters and we let $C\_MIN = -C\_MAX$.

4.3 Robustness Training Scheme

As we have obtained the theoretical guarantee between the received messages and the learned joint message representation, now this subsection describes how to acquire a robust communication policy. Following the popular Centralized Training and Decentralized Execution (CTDE) paradigm [26, 33],

during the training phase, we use a state encoder (e.g., additional VAE [24]) to encode the state s into a latent space with parameter $\boldsymbol{\psi}$:

$$\mathcal{L}(\boldsymbol{\psi}) = -\mathbb{E}_{q_{\boldsymbol{\psi}_{\text{enc}}}(\boldsymbol{z}_{\text{st}}|s)}[\log p_{\boldsymbol{\psi}_{\text{dec}}}(s \mid z_{\text{st}})] + \text{KL}\left[q_{\boldsymbol{\psi}_{\text{enc}}}(\boldsymbol{z}_{\text{st}} \mid s), p(\boldsymbol{z}_{\text{st}})\right], \tag{13}$$

where the operators are similar to Eqn. 1. We can then apply any robust single-agent RL algorithm to achieve robustness in the latent space of state representation. If the robustness of state representation is guaranteed, we can also ensure the robustness of joint message representation through knowledge distillation mentioned in Eqn. 16. We here implement our method on RADIAL-RL [41] as it principles a framework in adversarial training with strong theoretical guarantee and robustness performance. Then we can minimize the following loss to optimize each agent's individual policy:

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{(s,\boldsymbol{a},s',r)}\left[\sum_i \sum_y Q^i_{\text{diff}}(\tau, z_{\text{st}}; y) \cdot Ovl^i(\tau, z_{\text{st}}, \kappa\epsilon; y)\right], \tag{14}$$

with

$$\begin{aligned} Q^i_{\text{diff}}(\tau, z_{\text{st}}; y) &= \max(0, Q^i(\tau, z_{\text{st}}; a) - Q^i(\tau, z_{\text{st}}; y)), \\ Ovl^i(\tau, z_{\text{st}}, \kappa\epsilon; y) &= \max(0, \overline{Q}^i(\tau, z_{\text{st}}, \kappa\epsilon; y) - \underline{Q}^i(\tau, z_{\text{st}}, \kappa\epsilon; a)), \end{aligned} \tag{15}$$

where i is the identification of each agent, y is each action, a is the chosen action, and $\overline{Q}, \underline{Q}$ can be computed by interval bound propagation under ℓ_∞-norm perturbation within budget $\kappa\epsilon$, which is readily available as there is only MLP network existing between the Q-values and (τ, z_{st}). Ovl represents the overlap between the bounds of two actions which can be seen in Fig. 1, and Q_{diff} measures the relative quality between two actions as we can ignore the overlap if they are similar enough. When minimizing the weighted overlap to 0, which means even the upper bound of another action y's action-value $\overline{Q}^i(\tau, z_{\text{st}}, \kappa\epsilon; y)$ is lower than the lower bound of original action a's action-value $\underline{Q}^i(\tau, z_{\text{st}}, \kappa\epsilon; a)$, the agent won't change its action under perturbation, leading to a robust communication policy. We note that the model's initial training will be hindered if we add the robust loss. Therefore, it is better to start robust training after the training is stable, and we use T_r to control it. Then we optimize the joint message representation by minimizing the KL divergence between z_{st} and z_{msg} as a form of knowledge distillation, and we use only the message encoder $\boldsymbol{\phi}_{\text{enc}}$ to make the inference of the joint message representation:

$$\mathcal{L}(\boldsymbol{\phi}) = \text{KL}\left[sg(z_{\text{st}}), q_{\boldsymbol{\phi}_{\text{enc}}}(z_{\text{msg}} \mid m)\right], \tag{16}$$

where $sg(\cdot)$ denotes gradient stop, and z_{msg} is the joint message representations sampled from $\mathcal{N}(\mu_i, \sigma_i^2)$. Then the overall loss function is as follows:

$$\mathcal{L} = \mathcal{L}_{\text{TD}}(\boldsymbol{\theta}) + \alpha_1\mathcal{L}(\boldsymbol{\psi}) + \alpha_2\mathcal{L}(\boldsymbol{\phi}) + \mathbb{I}(t > T_r)\alpha_3\mathcal{L}_{\text{adv}}. \tag{17}$$

Here, $\mathcal{L}_{\text{TD}}(\boldsymbol{\theta})$ is the temporal difference loss, α_1, α_2, and α_3 are adjustable hyperparameters for each loss function accordingly, and $\mathbb{I}(\cdot)$ is the indicator function. In the CTDE framework, the mixing network will be removed during the decentralized execution phase. To prevent the lazy-agent problem [57] and reduce model complexity, we make the local network have the same parameters for all agents. The pseudo-code is shown in Appendix B.

5 Experimental Results

In this section, we design experiments on multiple complex scenarios to evaluate the communication robustness for the following questions: (1) How is the robustness of our proposed method on multiple benchmarks compared with baselines (Sec. 5.2) ? (2) What is the generalization ability of CroMAC when

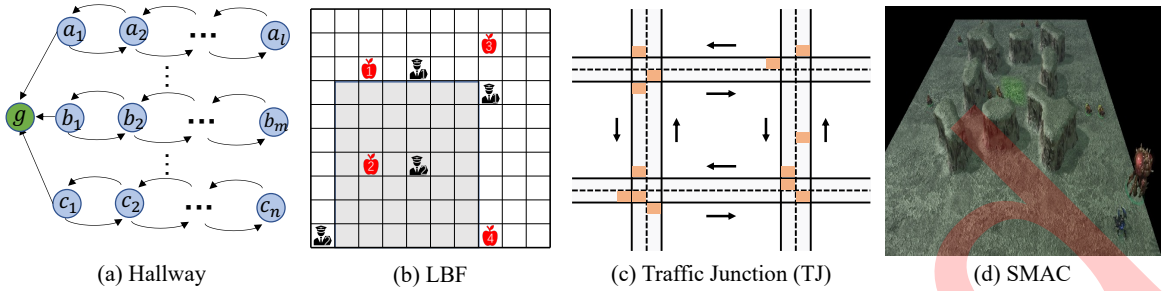


Figure 2 Multiple benchmarks used in our experiments.

encountering different message perturbations (Sec. 5.3) ? (3) Can CroMAC be integrated into multiple cooperative MARL methods in different communication conditions, and how does each hyperparameter influence its performance (Sec. 6) ?

For empirical evaluation, we compare CroMAC with multiple baselines on different cooperative tasks, including Hallway [66], Level-Based Foraging (LBF) [45], Traffic Junction (TJ) [9], and two maps from StarCraft Multi-Agent Challenge (SMAC) [66]. CroMAC is implemented on QMIX if not specified based on PyMARL[1)]. All results are illustrated with mean performance and standard error on 5 random seeds. Detailed network architecture and hyperparameter choices are shown in Appendix D.

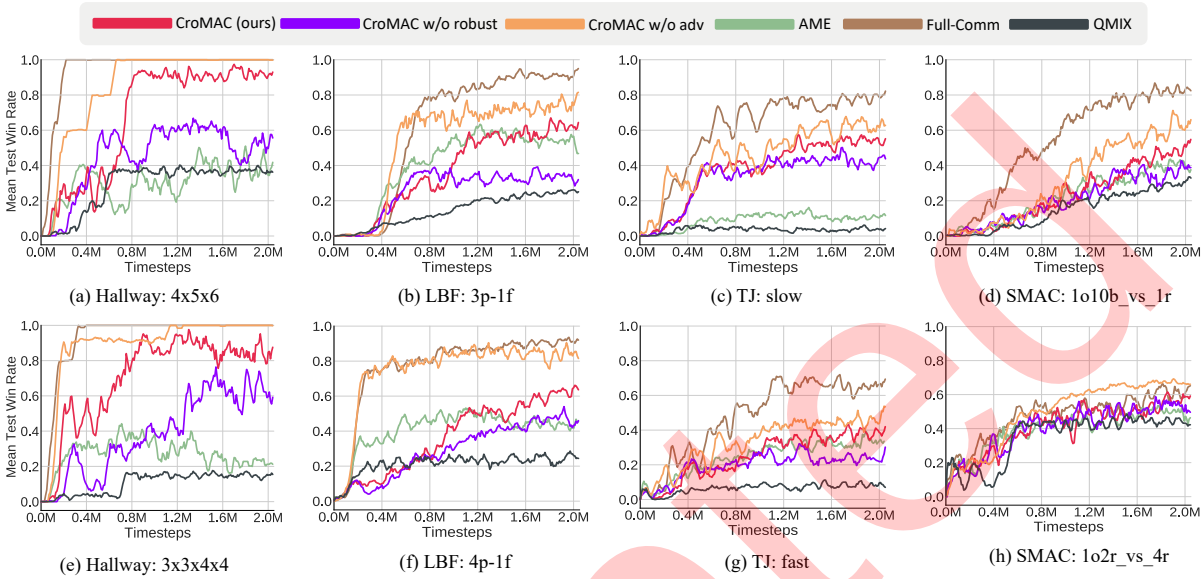
5.1 Baselines and Environments

We consider multiple baselines with different communication abilities, where QMIX [50] is a value-based baseline, and no message sharing among agents, showing excellent performance on diverse multi-agent benchmarks [45]. AME [55] is a recently proposed strong method for the robustness of multi-agent communication, which assumes no more than half of the agents may suffer from message perturbations, and an ensemble-based defense approach is then introduced to realize the robustness goal. Full-Comm adopts a full communication paradigm, where each agent receives messages from all teammates at each timestep without message perturbations both in the training and testing phases, which can be seen as an upper-bound performance algorithm. For the ablation studies, we consider multiple variants of CroMAC. CroMAC w/o robust and CroMAC w/o adv are two variants of our proposed CroMAC, the former does not have the proposed robust training scheme while the latter is conducted in the non-perturbed condition in both the training and testing phases. We consider multiple benchmarks, where Hallway [66] is a cooperative environment under partial observability, with m agents are randomly initialized at different positions and required to arrive at the goal g simultaneously. We consider two scenarios with different numbers of agents and lengths of the hallway. LBF [45] is another cooperative, partially observable grid world game where agents should coordinate to collect food concurrently. As the original version focuses on exploration, here we modify it by making only one agent able to observe the map, which needs strong communication to complete this task. TJ [9] is another popular benchmark used to test communication ability, where multiple cars move along two-way roads with one or more road junctions following predefined routes, and they need to drive as fast as possible while avoiding collisions. We test on the modified slow and fast scenarios where different maps have a different probability of adding new cars. Two maps named 1o_2r_vs_4r and 1o_10b_vs_1r from SMAC [66] that require efficient communication are also used to test the robustness in more complex scenarios. Details are presented in Appendix C.

5.2 Robustness Comparison and Analysis

We first compare CroMAC against multiple baselines to investigate the communication robustness on various benchmarks. As shown in Fig. 3, QMIX achieves the most inferior performance in all environments, demonstrating that communication is needed. Full-Comm can solve all the tasks under perturbation-free

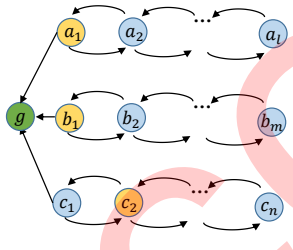
1) https://github.com/oxwhirl/pymarl



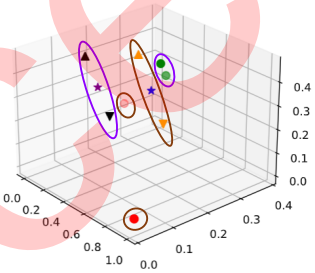
(a) Hallway: 4x5x6 (b) LBF: 3p-1f (c) TJ: slow (d) SMAC: 1o10b_vs_1r

(e) Hallway: 3x3x4x4 (f) LBF: 4p-1f (g) TJ: fast (h) SMAC: 1o2r_vs_4r

Figure 3 Empirical Results of several algorithms tested in two different perturbation conditions on benchmarks. Note that Full-Comm, CroMAC w/o adv, and QMIX are tested in perturbation-free conditions, while CroMAC, CroMAC w/o robust, and AME suffer from message perturbations when testing.



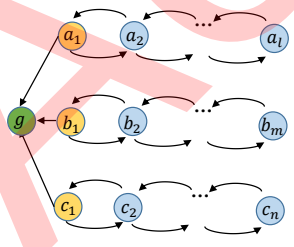
(a) t=3



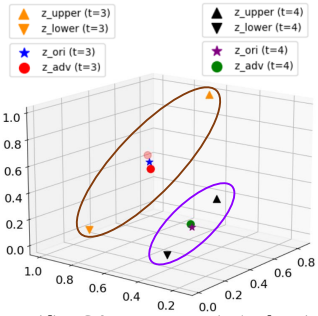
(c) PCA projection (w/o robust)

	still	left	right		still	left	right
Agent 1	0.22	-924.43	-11.52	Agent 1	-548.97	0.39	-159.88
	0.23	-505.67	1.78		0.24	-322.72	-12.10
	1.04	2.40	21.67		0.54	0.64	14.30
	-13.72	-3984.29	-89.09		-1208.64	-544.16	-3523.17
Agent 2	0.15	-829.25	-11.23	Agent 2	-603.77	0.38	-175.25
	0.21	-418.33	2.88		0.19	-320.74	-12.13
	0.91	4.16	16.60		-10.18	0.69	-41.10
	-12.73	-3886.47	-87.82		-1215.66	-529.83	-3542.15
Agent 3	-17.46	-3.50	-35.85	Agent 3	-617.34	0.39	-174.57
	-12.35	-2.94	-31.08		0.06	-225.45	-12.46
	13.05	-1.33	9.47		-56.64	0.74	-112.33
	-33.83	-5.80	-66.71		-1219.98	-412.51	-3558.12

(e) Q-values (w/o robust)



(b) t=4



(d) PCA projection (w/ robust)

	still	left	right		still	left	right
Agent 1	-1.79	-37.59	-2.60	Agent 1	-73.13	-8.02	-1371.43
	-1.77	-24.69	-2.85		40.38	-8.05	-877.14
	-1.76	-12.43	-2.12		-19.43	-6.63	-513.75
	-1.81	-62.74	-3.08		-129.83	-15.48	-2228.88
Agent 2	-1.78	-34.59	-2.60	Agent 2	-76.88	-7.97	-1409.60
	-1.77	-21.79	-2.90		-43.98	-8.01	-913.24
	-1.76	-9.44	-2.18		-21.16	-6.66	-563.92
	-1.81	-59.75	-3.14		-132.60	-17.39	-2255.06
Agent 3	-11.26	-8.44	-15.79	Agent 3	-90.14	-8.04	-1499.88
	-10.66	-8.47	-15.23		-57.26	-8.08	-1004.16
	-9.88	-8.28	-14.03		-32.39	-6.61	-601.60
	-12.95	-8.56	-19.59		-147.89	-16.27	-2398.04

(f) Q-values (w/ robust)

Figure 4 Visualization results. We take $t = 3$ and 4 in Hallway as shown in (a) and (b), where Agent 1 and Agent 2 stand one step from the goal while Agent 3 needs to take two steps to reach the goal. (c) and (d) show the PCA projection [59] of the message representation z_{msg} for (a) and (b), with $\bullet$ and $\star$ represent z_{msg} with and without perturbations, respectively. Note that z_{msg} is the same for agents without perturbations, and some $\bullet$ are darker because multiple ones overlap together. $\blacktriangle, \blacktriangledown$ represents the upper and lower bounds of z_{msg}, note that ellipses of the same color represent the same time step. (e) and (f) display the Q-values (multiplied by 100 for viewing) of each agent accordingly, where the first row means the original Q-values of all actions, while the second row refers to them under perturbation with red fonts representing the selected actions in corresponding cases. The third and fourth rows show the upper and lower bounds of Q-values under ϵ-perturbation, where yellow squares are the lower bounds of Q-values over best actions while blue squares are the upper bounds of Q-values over other actions.

https://engine.scichina.com/doi/10.1007/s11432-023-3853-y

Table 1 Average test win rates for CroMAC and AME under different message perturbation conditions. The results are averaged from 1000 test episodes among 5 random seeds, where FGSM (n) refers to methods under FGSM attack with different budgets. The details of each perturbation method are shown in Appendix D.

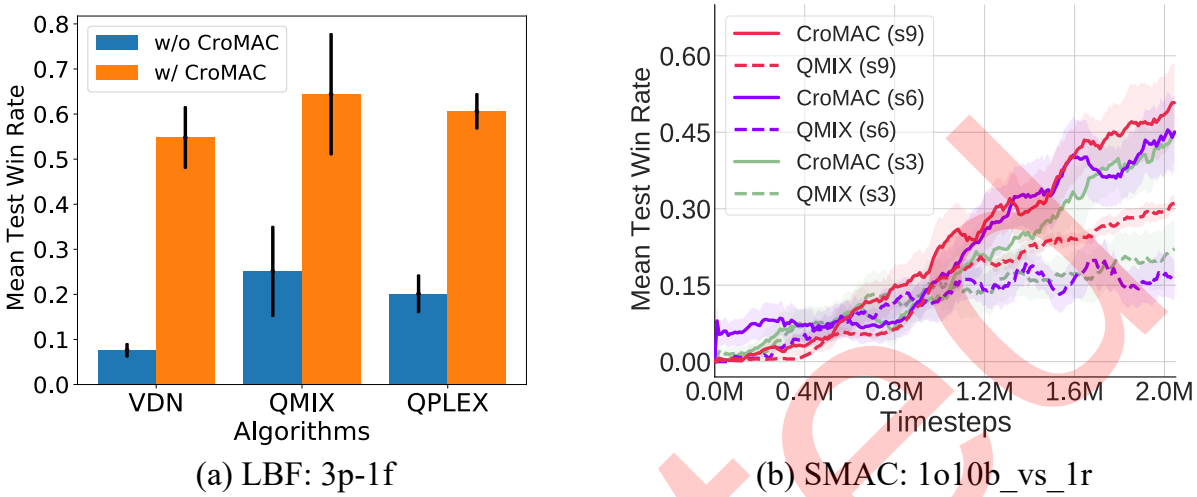
Environment	Method	Natural	Random	PGD	FGSM (1)	FGSM (2)	FGSM	FGSM (3)	FGSM (4)
Hallway	CroMAC	0.93±0.06	0.91±0.11	**0.92±0.13**	**0.97±0.03**	**0.86±0.04**	**0.91±0.10**	0.60±0.31	**0.66±0.39**
4x5x6	AME	0.98±0.01	0.93±0.04	0.43±0.20	0.66±0.34	0.61±0.34	0.62±0.31	0.36±0.10	0.10±0.20
	REC	**1.00±0.00**	**0.95±0.08**	0.90±0.20	0.96±0.06	0.62±0.38	0.82±0.23	**0.68±0.40**	0.41±0.43
LBF	CroMAC	0.71±0.05	0.72±0.03	**0.61±0.09**	**0.71±0.07**	**0.67±0.09**	**0.64±0.13**	**0.43±0.15**	**0.30±0.08**
3p-1f	AME	**0.77±0.04**	0.72±0.04	0.58±0.11	0.63±0.09	0.56±0.02	0.47±0.03	0.36±0.10	0.29±0.04
TJ	CroMAC	**0.31±0.07**	**0.46±0.20**	**0.31±0.23**	**0.29±0.12**	**0.31±0.09**	**0.37±0.14**	**0.42±0.16**	**0.32±0.18**
slow	AME	0.12±0.07	0.13±0.03	0.15±0.06	0.13±0.06	0.13±0.06	0.12±0.06	0.08±0.02	0.13±0.06
SMAC	CroMAC	**0.65±0.10**	**0.64±0.12**	**0.53±0.07**	**0.56±0.04**	**0.52±0.18**	**0.59±0.08**	0.41±0.14	0.34±0.03
1o10b_vs_1r	AME	0.38±0.12	0.52±0.24	0.51±0.17	0.44±0.13	0.45±0.20	0.38±0.02	**0.44±0.19**	**0.43±0.07**

conditions, showing that these tasks need communication and can be solved by a simple communication mechanism. CroMAC w/o adv, an ablation of CroMAC where testing is conducted under perturbation-free conditions, can achieve comparable coordination ability with Full-Comm, validating the specific design of our CroMAC does not cause much performance degradation for a communication goal. On the contrary, when message perturbations occur during the testing phase, it can be easily found that CroMAC w/o robust, a variant of our proposed method without an efficient robust mechanism, suffers from severe performance degradation compared with Full-Comm and CroMAC w/o adv. However, CroMAC exhibits higher robustness than others, and it surprises us that AME also suffers from severe performance degradation under perturbation, which means an unreasonable constraint for robustness training cannot be applied in complex and severe message perturbation conditions.

Furthermore, we conduct experiments on task Hallway to investigate how CroMAC learns a robust communication policy. As shown in Fig. 4, three agents coordinate to reach the goal. When suffering from perturbations, the message representation learned by methods without a robust mechanism will go out of the upper and lower bounds, leading to an unpredictable input for the local policy. Consequently, the message perturbation influences the action selection of each agent. Take Agent 1 in Fig. 4(e) as an example. It should keep still with Agent 2 at $t = 3$ to wait for Agent 3 to go left together for success. However, when suffering from perturbations, the message representation jumps out of the normal range. It unexpectedly goes right, as the according Q-value 1.78 is dominant to others (0.23 for still and -505.67 for left). On the contrary, with our robustness scheme, the message representations can be bounded in a reasonable range, leading to a robust action selection compared with the perturbation-free setting, as shown in Fig. 4(f). The whole process shows our approach can obtain message certification when any perturbations happen.

5.3 Robustness Under Various Perturbations

As this study considers a setting where the number of attacks is fixed during the training phase, we evaluate here the generalization ability when altering the perturbation budget and encountering different perturbation methods in the testing phase. Specifically, we conduct experiments on each benchmark with the same structures and hyperparameters as Sec. 5.2 during training. As shown in Tab. 1, we consider eight communication situations, where "Natural" means no perturbation exists, FGSM is the training condition of the comparable approaches, and others like PGD are other conditions, details can be seen in Appendix D. We can find that AME can achieve comparable or even superiority over CroMAC in the Natural setting without message perturbations and also maintain competitiveness when suffering from random perturbations, showing that AME possesses robustness for simple message perturbations. However, AME sustains a drastic performance degradation when we alter the perturbation budget like FGSM (4) or other perturbation models like PGD in the Hallway environment. On the other hand, our CroMAC achieves high superiority over AME in most environments under different perturbations, demonstrating its



(a) LBF: 3p-1f (b) SMAC: 1o10b_vs_1r

Figure 5 (a) Average test success rates of CroMAC implemented with different value based MARL methods. (b) Performance comparison with varying sights, where *sn* means the sight range is n and the default sight range is 9.

high generalization ability when encountering different perturbation budgets and perturbation methods.

5.4 Generality and Parameter Sensitivity

CroMAC is a robust communication training paradigm and is agnostic to specific value decomposition MARL methods and any sight conditions. Here we treat it as a plug-in module and integrate it with existing MARL value decomposition methods like VDN [57], QMIX [50], and QPLEX [64]. As shown in Fig. 5(a), when integrating with CroMAC, the performance of the baselines vastly improves on scenario LBF:3p-1f with message perturbations, indicating that the proposed training paradigm can significantly enhance robustness for various MARL methods. It is worth mentioning that QPLEX shows instability when adding robust loss and we choose the best result in the training process for comparison. Furthermore, when we alter the agents' sight range in the SMAC map 1o10b_vs_1r, the results shown in Fig. 5(b) also demonstrate that CroMAC can improve the coordination robustness in different sight conditions for QMIX with communication, showing its high generality under different communication scenarios.

As CroMAC includes multiple hyperparameters, here we conduct experiments on scenario Hallway: 4x5x6 to investigate how each one influences the robustness. Where κ controls the attack strength added to the latent state space. If it is too small, we cannot guarantee good robustness in the testing phase, and if it is too large, the policy may be too smooth and not optimal anymore. As shown in Fig. 6(a), we can find that $\kappa = 5$ is the best choice in this scenario. Furthermore, the value range of the network weight W is used to get an approximate bound of the integration error. Fig. 6(b) shows that $C\_MAX = 0.1$ performs best. We can find the most appropriate parameters for other scenarios in the same way. More details for other scenarios are shown in Appendix D. Besides, as there are multiple hyperparameters of each loss function, we show how each adjustable hyperparameter named α_1, α_2, and α_3 influence the robustness of CroMAC, we continue to conduct experiments on the task Hallway: 4x5x6 to investigate how each hyperparameter α influences the robustness. As shown in Fig. 6(c)-(e), we can find that when the parameter is slightly larger or smaller, the performance may suffer corresponding degradation and the stability will also decline.

6 Conclusion and Future Work

Considering the great significance of robustness for real-world policy deployment and the enormous potential of MARL, this paper takes a further step towards robustness in MARL communication. We first model the multi-agent communication as a multi-view problem and apply a multi-view variational au-

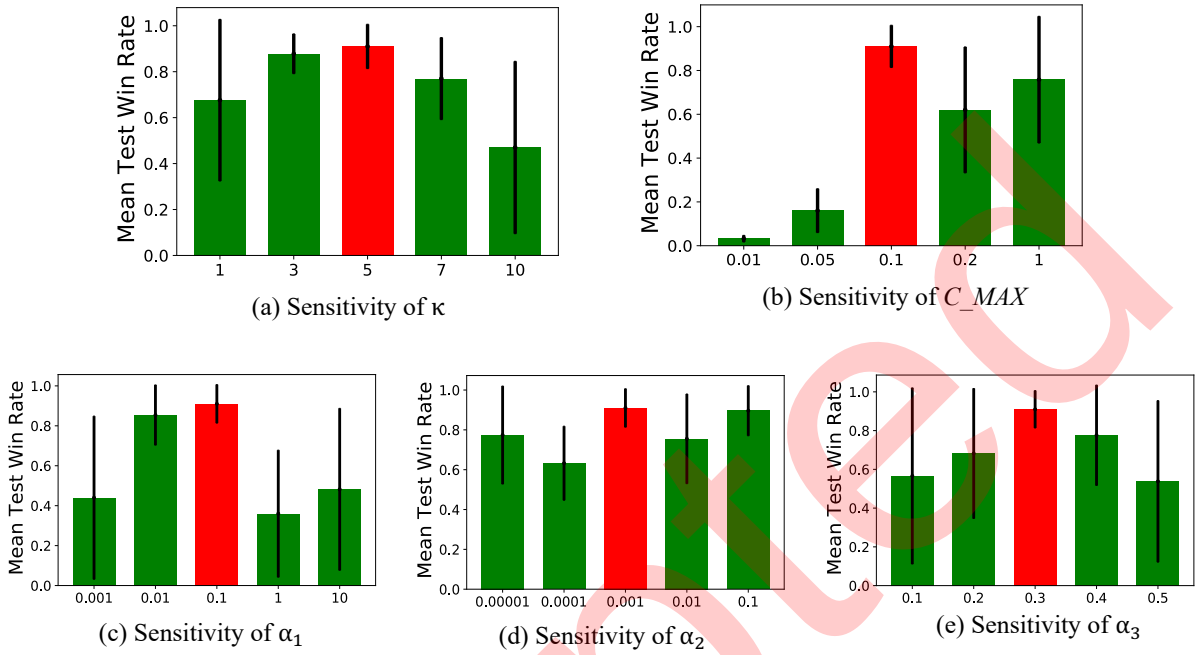


Figure 6 Sensitivity of hyperparameters used in this paper.

toencoder that uses a product-of-experts inference network to obtain a joint message representation from the received messages, then a certificate guarantee between the joint message representation and each received message is obtained via interval bound propagation. For the optimization phase, we first encode the state into a latent space, and do perturbations in this space to get a certificate state representation. Then the learned joint message representation is used to approximate the certificate state representation. Extensive experimental results from multiple aspects demonstrate the efficiency of the proposed method. In terms of possible future work, as we learn the communication policy online, how we can learn a robust communication policy in offline MARL is challenging but of great value.

Acknowledgements This work is supported by the National Key Research and Development Program of China (2020AAA0107200), the National Science Foundation of China (61921006, 61876119, 62276126), the Natural Science Foundation of Jiangsu (BK20221442), and the program B for Outstanding PhD candidate of Nanjing University. We thank Ziqian Zhang and Fuxiang Zhang for their useful suggestions.

References

1 Khaled Bayoudh, Raja Knani, Fayçal Hamdaoui, and Abdellatif Mtibaa. 2022. A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets. *The Visual Computer* 38, 8 (2022), 2939–2970.

2 Jan Blumenkamp and Amanda Prorok. 2021. The Emergence of Adversarial Communication in Multi-Agent Reinforcement Learning. In *CoRL*. 1394–1414.

3 Lucian Busoniu, Robert Babuska, and Bart De Schutter. 2008. A Comprehensive Survey of Multiagent Reinforcement Learnin. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38, 2 (2008), 156–172.

4 Jiahan Cao, Lei Yuan, Jianhao Wang, Shaowei Zhang, Chongjie Zhang, Yang Yu, and De-Chuan Zhan. 2021. LINDA: Multi-Agent Local Information Decomposition for Awareness of Teammates. *arXiv preprint arXiv:2109.12508* (2021).

5 Yanshuai Cao and David J Fleet. 2014. Generalized Product of Experts for Automatic and Principled Fusion of Gaussian Process Predictions. *arXiv preprint arXiv:1410.7827* (2014).

6 Ning Chen, Jun Zhu, Fuchun Sun, and Eric Poe Xing. 2012. Large-Margin Predictive Latent Subspace Learning for Multiview Data Analysis. *IEEE transactions on pattern analysis and machine intelligence* 34, 12 (2012), 2365–2378.

7 Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*. 1724–1734.

8 Filippos Christianos, Georgios Papoudakis, Muhammad A Rahman, and Stefano V Albrecht. 2021. Scaling Multi-Agent Reinforcement Learning with Selective Parameter Sharing. In *ICML*. 1989–1998.

9 Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. 2019. TarMAC:Targeted Multi-Agent Communication. In *ICML*. 1538–1546.

10 Ziluo Ding, Tiejun Huang, and Zongqing Lu. 2020. Learning Individually Inferred Communication for Multi-Agent Cooperation. In *NeurIPS*.

11 Ali Dorri, Salil S Kanhere, and Raja Jurdak. 2018. Multi-agent systems: A survey. *IEEE Access* 6 (2018), 28573–28593.

12 Michael Everett, Björn Lütjens, and Jonathan P. How. 2022. Certifiable Robustness to Adversarial State Uncertainty in Deep Reinforcement Learning. *IEEE Transactions on Neural Networks and Learning Systems* 33, 9 (2022), 4184–4198.

13 Jiameng Fan and Wenchao Li. 2022. DRIBO: Robust Deep Reinforcement Learning via Multi-View Information Bottleneck. In *ICML*. 6074–6102.

14 Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. 2016. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In *NeurIPS*. 2137–2145.

15 Walter Gautschi. 1974. A harmonic mean inequality for the gamma function. *SIAM Journal on Mathematical Analysis* 5, 2 (1974), 278–281.

16 Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *ICLR*.

17 Rihab Gorsane, Omayma Mahjoub, Ruan John de Kock, Roland Dubb, Siddarth Singh, and Arnu Pretorius. 2022. Towards a Standardised Performance Evaluation Protocol for Cooperative MARL. In *NeurIPS*.

18 Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. 2018. On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models. *arXiv preprint arXiv:1810.12715* (2018).

19 Sven Gronauer and Klaus Diepold. 2022. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review* 55, 2 (2022), 895–943.

20 Jun Guo, Yonghong Chen, Yihang Hao, Zixin Yin, Yin Yu, and Simin Li. 2022. Towards Comprehensive Testing on the Robustness of Cooperative Multi-agent Reinforcement Learning. *arXiv preprint arXiv:2204.07932* (2022).

21 Yizheng Hu, Kun Shao, Dong Li, Jianye Hao, Wulong Liu, Yaodong Yang, Jun Wang, and Zhanxing Zhu. 2021. Robust Multi-Agent Reinforcement Learning Driven by Correlated Equilibrium. https://openreview.net/forum?id=JvPsKam58LX

22 Yizheng Hu and Zhihua Zhang. 2022. Sparse Adversarial Attack in Multi-agent Reinforcement Learning. *arXiv preprint arXiv:2205.09362* (2022).

23 HyeongJoo Hwang, Geon-Hyeong Kim, Seunghoon Hong, and Kee-Eung Kim. 2021. Multi-View Representation Learning via Total Correlation Objective. In *NeurIPS*. 12194–12207.

24 Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *ICLR*.

25 Akira Kinose, Masashi Okada, Ryo Okumura, and Tadahiro Taniguchi. 2022. Multi-View Dreaming: Multi-View World Model with Contrastive Learning. *arXiv preprint arXiv:2203.11024* (2022).

26 Landon Kraemer and Bikramjit Banerjee. 2016. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing* 190 (2016), 82–94.

27 Minne Li, Lisheng Wu, Jun Wang, and Haitham Bou Ammar. 2019. Multi-View Reinforcement Learning. In *NeurIPS*. 1418–1429.

28 Shihui Li, Yi Wu, Xinyue Cui, Honghua Dong, Fei Fang, and Stuart Russell. 2019. Robust Multi-Agent Reinforcement Learning via Minimax Deep Deterministic Policy Gradient. In *AAAI*. 4213–4220.

29 Yingming Li, Ming Yang, and Zhongfei Zhang. 2018. A Survey of Multi-View Representation Learning. *IEEE transactions on knowledge and data engineering* 31, 10 (2018), 1863–1883.

30 Yongyuan Liang, Yanchao Sun, Ruijie Zheng, and Furong Huang. 2022. Efficient Adversarial Training without Attacking: Worst-Case-Aware Robust Reinforcement Learning. In *NeurIPS*.

31 Jieyu Lin, Kristina Dzeparoska, Sai Qian Zhang, Alberto Leon-Garcia, and Nicolas Papernot. 2020. On the Robustness of Cooperative Multi-Agent Reinforcement Learning. In *SPW*. 62–68.

32 Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb Pieter Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *NeurIPS*. 6379–6390.

33 Xueguang Lyu, Yuchen Xiao, Brett Daley, and Christopher Amato. 2021. Contrasting Centralized and Decentralized Critics in Multi-Agent Reinforcement Learning. In *AAMAS*. 844–852.

34 Mengmeng Ma, Jian Ren, Long Zhao, Sergey Tulyakov, Cathy Wu, and Xi Peng. 2021. SMIL: Multimodal Learning with Severely Missing Modality. In *AAAI*. 2302–2310.

35 David J. C. MacKay. 2003. *Information theory, inference, and learning algorithms*. Cambridge University Press.

36 Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*.

37 Hangyu Mao, Zhengchao Zhang, Zhen Xiao, Zhibo Gong, and Yan Ni. 2020. Learning Agent Communication under Limited Bandwidth by Message Pruning. In *AAAI*. 5142–5149.

38 Hangyu Mao, Zhengchao Zhang, Zhen Xiao, Zhibo Gong, and Yan Ni. 2020. Learning multi-agent communication with double attentional deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems* 34, 1 (2020), 1–34.

39 Rupert Mitchell, Jan Blumenkamp, and Amanda Prorok. 2020. Gaussian Process Based Message Filtering for Robust Multi-Agent Cooperation in the Presence of Adversarial Communication. *arXiv preprint arXiv:2012.00508* (2020).

40 Janosch Moos, Kay Hansel, Hany Abdulsamad, Svenja Stark, Debora Clever, and Jan Peters. 2022. Robust Reinforcement Learning: A Review of Foundations and Recent Advances. *Machine Learning and Knowledge Extraction* 4, 1 (2022), 276–315.

41 Tuomas Oikarinen, Wang Zhang, Alexandre Megretski, Luca Daniel, and Tsui-Wei Weng. 2021. Robust Deep Reinforcement Learning through Adversarial Loss. In *NeurIPS*. 26156–26167.

42 Afshin OroojlooyJadid and Davood Hajinezhad. 2019. A Review of Cooperative Multi-Agent Deep Reinforcement Learning. *arXiv preprint arXiv:1908.03963* (2019).

43 Xinlei Pan, Daniel Seita, Yang Gao, and John Canny. 2019. Risk Averse Robust Adversarial Reinforcement Learning. In *ICRA*. 8522–8528.

44 Georgios Papoudakis, Filippos Christianos, Arrasy Rahman, and Stefano V Albrecht. 2019. Dealing with Non-Stationarity in Multi-Agent Deep Reinforcement Learning. *arXiv preprint arXiv:1906.04737* (2019).

45 Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V Albrecht. 2021. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative Tasks. In *NeurIPS*.

46 Hyunjong Park, Sanghoon Lee, Junghyup Lee, and Bumsub Ham. 2021. Learning by Aligning: Visible-Infrared Person Re-identification using Cross-Modal Correspondences. In *ICCV*. 12046–12055.

47 Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. 2017. Robust Adversarial Reinforcement Learning. In *ICML*. 2817–2826.

48 You Qiaoben, Chengyang Ying, Xinning Zhou, Hang Su, Jun Zhu, and Bo Zhang. 2021. Understanding Adversarial Attacks on Observations in Deep Reinforcement Learning. *arXiv preprint arXiv:2106.15860* (2021).

49 Zengyi Qin, Kaiqing Zhang, Yuxiao Chen, Jingkai Chen, and Chuchu Fan. 2020. Learning Safe Multi-agent Control with Decentralized Neural Barrier Certificates. In *ICLR*.

50 Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *ICML*. 4295–4304.

51 Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning Structured Output Representation using Deep Conditional Generative Models.. In *NeurIPS*. 3483–3491.

52 Yeeho Song and Jeff Schneider. 2022. Robust Reinforcement Learning via Genetic Curriculum. In *ICRA*. 5560–5566.

53 Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. 2016. Learning Multiagent Communication with Backpropagation. In *NeurIPS*. 2244–2252.

54 Chuangchuang Sun, Dong-Ki Kim, and Jonathan P How. 2022. ROMAX: Certifiably Robust Deep Multiagent Reinforcement Learning via Convex Relaxation. In *ICRA*. 5503–5510.

55 Yanchao Sun, Ruijie Zheng, Parisa Hassanzadeh, Yongyuan Liang, Soheil Feizi, Sumitra Ganesh, and Furong Huang. 2022. Certifiably Robust Policy Learning against Adversarial Communication in Multi-agent Systems. *arXiv preprint arXiv:2206.10158* (2022).

56 Yanchao Sun, Ruijie Zheng, Yongyuan Liang, and Furong Huang. 2021. Who Is the Strongest Enemy? Towards Optimal and Efficient Evasion Attacks in Deep RL. In *ICLR*.

57 Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinícius Flores Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, and Thore Graepel. 2018. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *AAMAS*. 2085–2087.

58 Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. 2016. Joint Multimodal Learning with Deep Generative Models. *arXiv preprint arXiv:1611.01891* (2016).

59 Michael E Tipping and Christopher M Bishop. 1999. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61, 3 (1999), 611–622.

60 James Tu, Tsunhsuan Wang, Jingkang Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. 2021. Adversarial Attacks On Multi-Agent Communication. In *ICCV*. 7768–7777.

61 Tessa van der Heiden, Christoph Salge, Efstratios Gavves, and Herke van Hoof. 2020. Robust Multi-Agent Reinforcement Learning with Social Empowerment for Coordination and Communication. *arXiv preprint arXiv:2012.08255* (2020).

62 Eugene Vinitsky, Yuqing Du, Kanaad Parvate, Kathy Jang, Pieter Abbeel, and Alexandre Bayen. 2020. Robust Reinforcement Learning using Adversarial Populations. *arXiv preprint arXiv:2008.01825* (2020).

63 Jianhao Wang, Zhizhou Ren, Beining Han, Jianing Ye, and Chongjie Zhang. 2021. Towards Understanding Cooperative Multi-Agent Q-Learning with Value Factorization. In *NeurIPS*. 29142–29155.

64 Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. 2021. QPLEX: Duplex Dueling Multi-Agent Q-Learning. In *ICLR*.

65 Jianhong Wang, Wangkun Xu, Yunjie Gu, Wenbin Song, and Tim C Green. 2021. Multi-Agent Reinforcement Learning for Active Voltage Control on Power Distribution Networks. In *NeurIPS*. 3271–3284.

66 Tonghan Wang, Jianhao Wang, Chongyi Zheng, and Chongjie Zhang. 2020. Learning Nearly Decomposable Value Functions Via Communication Minimization. In *ICLR*.

67 Yihan Wang, Beining Han, Tonghan Wang, Heng Dong, and Chongjie Zhang. 2021. DOP: Off-Policy Multi-Agent Decomposed Policy Gradients. In *ICLR*.

68 Yuanfei Wang, Jing Xu, Yizhou Wang, et al. 2021. ToM2C: Target-oriented Multi-agent Communication and Cooperation with Theory of Mind. In *ICLR*.

69 Muning Wen, Jakub Grudzien Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. 2022. Multi-Agent Reinforcement Learning is a Sequence Modeling Problem. In *NeurIPS*.

70 Fan Wu, Linyi Li, Zijian Huang, Yevgeniy Vorobeychik, Ding Zhao, and Bo Li. 2021. CROP: Certifying Robust Policies for Reinforcement Learning through Functional Smoothing. In *ICLR*.

71 Fan Wu, Linyi Li, Huan Zhang, Bhavya Kailkhura, Krishnaram Kenthapadi, Ding Zhao, and Bo Li. 2021. COPA: Certifying Robust Policies for Offline Reinforcement Learning against Poisoning Attacks. In *ICLR*.

72 Junlin Wu and Yevgeniy Vorobeychik. 2022. Robust Deep Reinforcement Learning through Bootstrapped Opportunistic Curriculum. In *ICML*. 24177–24211.

73 Mike Wu and Noah Goodman. 2018. Multimodal Generative Models for Scalable Weakly-Supervised Learning. In *NeurIPS*. 5580–5590.

74 Chang Xu, Dacheng Tao, and Chao Xu. 2015. Multi-View Learning With Incomplete Views. *IEEE Transactions on Image Processing* 24, 12 (2015), 5812–5825.

75 Jinglin Xu, Wenbin Li, Xinwang Liu, Dingwen Zhang, Ji Liu, and Junwei Han. 2020. Deep Embedded Complementary and Interactive Information for Multi-View Classification. In *AAAI*. 6494–6501.

76 Mengdi Xu, Zuxin Liu, Peide Huang, Wenhao Ding, Zhepeng Cen, Bo Li, and Ding Zhao. 2022. Trustworthy Reinforcement Learning Against Intrinsic Vulnerabilities: Robustness, Safety, and Generalizability. *arXiv preprint arXiv:2209.08025* (2022).

77 Di Xue, Lei Yuan, Zongzhang Zhang, and Yang Yu. 2022. Efficient Multi-Agent Communication via Shapley Message Value. In *IJCAI*. 578–584.

78 Ke Xue, Jiacheng Xu, Lei Yuan, Miqing Li, Chao Qian, Zongzhang Zhang, and Yang Yu. 2022. Multi-agent Dynamic Algorithm Configuration. In *NeurIPS*.

79 Wanqi Xue, Wei Qiu, Bo An, Zinovi Rabinovich, Svetlana Obraztsova, and Chai Kiat Yeo. 2022. Mis-spoke or mis-lead: Achieving Robustness in Multi-Agent Communicative Reinforcement Learning. In *AAMAS*. 1418–1426.

80 Xiaoqiang Yan, Shizhe Hu, Yiqiao Mao, Yangdong Ye, and Hui Yu. 2021. Deep multi-view learning methods: a review. *Neurocomputing* 448 (2021), 106–129.

81 Jianing Ye, Chenghao Li, Jianhao Wang, and Chongjie Zhang. 2022. Towards Global Optimality in Cooperative MARL with Sequential Transformation. *arXiv preprint arXiv:2207.11143* (2022).

82 Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. 2021. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. *arXiv preprint arXiv:2103.01955* (2021).

83 Jing Yu, Clement Gehring, Florian Schäfer, and Animashree Anandkumar. 2021. Robust Reinforcement Learning: A Constrained Game-theoretic Approach. In *L4DC*. 1242–1254.

84 Lei Yuan, Chenghe Wang, Jianhao Wang, Fuxiang Zhang, Feng Chen, Cong Guan, Zongzhang Zhang, Chongjie Zhang, and Yang Yu. 2022. Multi-Agent Concentrative Coordination with Decentralized Task Representation. In *IJCAI*. 599–605.

85 Lei Yuan, Jianhao Wang, Fuxiang Zhang, Chenghe Wang, Zongzhang Zhang, Yang Yu, and Chongjie Zhang. 2022. Multi-Agent Incentive Communication via Decentralized Teammate Modeling. In *AAAI*. 9466–9474.

86 Won Joon Yun, Soohyun Park, Joongheon Kim, Myungjae Shin, Soyi Jung, Aziz Mohaisen, and Jae-Hyun Kim. 2022. Cooperative Multi-Agent Deep Reinforcement Learning for Reliable Surveillance via Autonomous Multi-UAV Control. *IEEE Transactions on Industrial Informatics* (2022).

87 Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. 2020. Robust Deep Reinforcement Learning against Adversarial Perturbations on State Observations. In *NeurIPS*. 21024–21037.

88 Kaiqing Zhang, Tao Sun, Yunzhe Tao, Sahika Genc, Sunil Mallya, and Tamer Basar. 2020. Robust Multi-Agent Reinforcement Learning with Model Uncertainty. In *NeurIPS*. 10571–10583.

89 Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. 2021. Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. *Handbook of Reinforcement Learning and Control* (2021), 321–384.

90 Sai Qian Zhang, Qi Zhang, and Jieyu Lin. 2019. Efficient Communication in Multi-Agent Reinforcement Learning via Variance Based Control. In *NeurIPS*. 3230–3239.

91 Sai Qian Zhang, Qi Zhang, and Jieyu Lin. 2020. Succinct and Robust Multi-Agent Communication With Temporal Message Control. In *NeurIPS*. 17271–17282.

92 Ziyuan Zhou and Guanjun Liu. 2022. RoMFAC: A Robust Mean-Field Actor-Critic Reinforcement Learning against Adversarial Perturbations on States. *arXiv preprint arXiv:2205.07229* (2022).

93 Changxi Zhu, Mehdi Dastani, and Shihan Wang. 2022. A Survey of Multi-Agent Reinforcement Learning with Communication. *arXiv preprint arXiv:2203.08975* (2022).

Appendix A Product of a Finite Number of Gaussians

Suppose we have N Gaussian experts with means $\mu_{i1}, \mu_{i2}, \cdots, \mu_{iN}$ and variances $\sigma_{i1}^2, \sigma_{i2}^2, \cdots, \sigma_{iN}^2$, the product distribution is still Gaussian with mean μ_i and variance σ_i^2:

$$\begin{aligned} \mu_i &= \Big(\frac{\mu_{i1}}{\sigma_{i1}^2} + \frac{\mu_{i2}}{\sigma_{i2}^2} + \cdots + \frac{\mu_{iN}}{\sigma_{iN}^2}\Big)\sigma_i^2, \\ \frac{1}{\sigma_i^2} &= \frac{1}{\sigma_{i1}^2} + \frac{1}{\sigma_{i2}^2} + \cdots + \frac{1}{\sigma_{iN}^2}. \end{aligned} \tag{A1}$$

It can be proved by induction.

Proof.

We want to prove Eqn. A1 is true for all $N \geqslant 2$.

- Base case: Suppose $N = 2$ and $p_1(x) = \mathcal{N}(x|\mu_1, \sigma_1), p_2(x) = \mathcal{N}(x|\mu_2, \sigma_2)$, then

$$\begin{aligned} p_1(x)p_2(x) &= \frac{1}{\sqrt{2\pi}\sigma_1}\exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right)\cdot\frac{1}{\sqrt{2\pi}\sigma_2}\exp\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right) \\ &= \frac{1}{2\pi\sigma_1\sigma_2}\exp\left(-\left(\frac{(x-\mu_1)^2}{2\sigma_1^2} + \frac{(x-\mu_2)^2}{2\sigma_2^2}\right)\right) \\ &= \frac{1}{2\pi\sigma_1\sigma_2}\exp\left(-\frac{x^2 - 2\frac{\mu_1\sigma_2^2+\mu_2\sigma_1^2}{\sigma_1^2+\sigma_2^2}x + \frac{\mu_1^2\sigma_2^2+\mu_2^2\sigma_1^2}{\sigma_1^2+\sigma_2^2}}{2\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}}\right) \\ &= \frac{1}{2\pi\sigma_1\sigma_2}\exp\left(-\frac{\left(x - \frac{\mu_1\sigma_2^2+\mu_2\sigma_1^2}{\sigma_1^2+\sigma_2^2}\right)^2}{2\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}} - \frac{(\mu_1-\mu_2)^2}{2\sigma_1^2\sigma_2^2}\right) \\ &= \frac{\exp\left(-\frac{(\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}\right)}{\sqrt{2\pi\left(\sigma_1^2+\sigma_2^2\right)}}\cdot\frac{1}{\sqrt{2\pi}\frac{\sigma_1\sigma_2}{\sqrt{\sigma_1^2+\sigma_2^2}}}\exp\left(-\frac{\left(x - \frac{\mu_1\sigma_2^2+\mu_2\sigma_1^2}{\sigma_1^2+\sigma_2^2}\right)^2}{2\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}}\right) \\ &= A\cdot\frac{1}{\sqrt{2\pi}\frac{\sigma_1\sigma_2}{\sqrt{\sigma_1^2+\sigma_2^2}}}\exp\left(-\frac{\left(x - \frac{\mu_1\sigma_2^2+\mu_2\sigma_1^2}{\sigma_1^2+\sigma_2^2}\right)^2}{2\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}}\right). \end{aligned} \tag{A2}$$

Eqn. A2 can be seen as PDF of $\mathcal{N}(\mu, \sigma)$ times A where $\mu = (\frac{\mu_1}{\sigma_1^2} + \frac{\mu_2}{\sigma_2^2})\sigma^2$, $\frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}$.

- Induction step: Suppose it is true when $N = n$, and the product distribution of n Gaussian experts has mean $\tilde{\mu} = (\frac{\mu_1}{\sigma_1^2} + \cdots + \frac{\mu_n}{\sigma_n^2})\tilde{\sigma}^2$ and variance $\frac{1}{\tilde{\sigma}^2} = \frac{1}{\sigma_1^2} + \cdots + \frac{1}{\sigma_n^2}$, then for $n+1$ Gaussian experts:

$$\begin{aligned} \frac{1}{\sigma^2} &= \frac{1}{\tilde{\sigma}^2} + \frac{1}{\sigma_{n+1}^2} = \frac{1}{\sigma_1^2} + \cdots + \frac{1}{\sigma_n^2} + \frac{1}{\sigma_{n+1}^2}, \\ \mu &= \Big(\frac{\tilde{\mu}}{\tilde{\sigma}^2} + \frac{\mu_{n+1}}{\sigma_{n+1}^2}\Big)\sigma^2 = \Big(\frac{\mu_1}{\sigma_1^2} + \cdots + \frac{\mu_n}{\sigma_n^2} + \frac{\mu_{n+1}}{\sigma_{n+1}^2}\Big)\sigma^2. \end{aligned} \tag{A3}$$

- Eqn. A1 has been proved by the above derivation.

If we write $T_{ij} = (\sigma_{ij}^2)^{-1}$, then Eqn. A1 can be written as:

$$\begin{aligned} \mu_i &= \Big(\sum_{j=1}^N \mu_{ij}T_{ij}\Big)\Big(\sum_{j=1}^N T_{ij}\Big)^{-1}, \\ \sigma_i^2 &= \Big(\sum_{j=1}^N T_{ij}\Big)^{-1}, \end{aligned} \tag{A4}$$

and is exactly what we're trying to prove.

Appendix B Algorithm

The whole optimization process is shown in Alg. A1. Where lines 6 to 7 are used to train policy network with only TD-error such that it has nothing to do with robust training; lines 8 to 10 aim at encoding the state into a latent space, while lines 11 to 13 train the MVAE with only partial observation and the received messages for decentralized execution, where $clamp(input, min, max)$ means clamping all elements in $input$ into range $[min, max]$; we train the policy network to be robust with auxiliary loss from lines 14 to 17.

Algorithm A1 CroMAC

Input: env, ϵ, κ, C_MIN, C_MAX, α_1, α_2, α_3, T, T_r
Initialize: Randomly initialize $\boldsymbol{\theta}$, $\boldsymbol{\psi}_{\text{enc}}$, $\boldsymbol{\psi}_{\text{dec}}$, $\boldsymbol{\phi}_{\text{enc}}$, and initialize empty replay buffer D

$t = 0$
while $t < T$ **do**
  Collect trajectory h from env with $\boldsymbol{\theta}(\mathbf{a}|\phi_{\text{enc}}(\mathbf{s}))$ and update replay buffer D
  for $j = 1, 2, \cdots$ **do**
    Sample a batch of Episodes from D
    Update policy network:
      $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$
    Update VAE ψ:
      $\boldsymbol{\psi}_{\text{enc}} \leftarrow \boldsymbol{\psi}_{\text{enc}} - \alpha_1 \nabla_{\boldsymbol{\psi}_{\text{enc}}} (\mathcal{L}(\boldsymbol{\psi}) + \mathcal{L}(\boldsymbol{\theta}))$
      $\boldsymbol{\psi}_{\text{dec}} \leftarrow \boldsymbol{\psi}_{\text{dec}} - \alpha_1 \nabla_{\boldsymbol{\psi}_{\text{dec}}} \mathcal{L}(\boldsymbol{\psi})$
    Update MVAE ϕ:
      $\boldsymbol{\phi}_{\text{enc}} \leftarrow \boldsymbol{\phi}_{\text{enc}} - \alpha_2 \nabla_{\boldsymbol{\phi}_{\text{enc}}} \mathcal{L}_(\boldsymbol{\phi})$
      Clamp: $\boldsymbol{\phi}_{\text{enc}} \leftarrow clamp(\boldsymbol{\phi}_{\text{enc}}, C\_MIN, C\_MAX)$
    if $t > T_r$ **then**
      Update policy network:
        $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha_3 \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{adv}}$
    end if
    Update the target network $\boldsymbol{\theta}^-$ at regular intervals
  end for
  Update t
end while

Appendix C Details about Baselines and Benchmarks

We compare CroMAC against different baselines and variants on diverse multi-agent benchmarks, and we introduce more details about these baselines here.

AME. Ablated Message Ensemble (AME) is a recently proposed certifiable defense method for multi-agent communication, which can guarantee agents' robustness when only part of communication messages suffer from perturbations. Specifically, AME makes a mild assumption that the attacker can only manipulate no more than half of the communication messages and trains a message-ablation policy that takes in a subset of messages from other agents and outputs a base action. In deployment, an ensemble policy is introduced by aggregating multiple base actions from multiple subsets of messages, achieving the robustness goal to some extent. The pseudocode can be seen in Alg.C1 and Alg.C2.

Algorithm C1 AME in the training phase

Input: env, ablation size k
Initialize: Randomly initialize $\hat{\pi}_i$ for each agent i.

$t = 0$
while $t < T$ **do**
  for $i = 1$ to N **do**
    Receive a set of messages $\mathbf{m}_{:\to \mathrm{i}}$, and update local history τ_i
    Randomly sample a subset of messages $[\mathbf{m}_{:\to \mathrm{i}}]_{\mathrm{k}} \sim \text{Uniform}(\mathcal{H}_{\mathrm{k}}(\mathbf{m}_{:\to \mathrm{i}}))$
    Take action based on τ_i and the message subset $[\mathbf{m}_{:\to \mathrm{i}}]_{\mathrm{k}}$, i.e., $a_i \leftarrow \hat{\pi}_i(\tau_i, [\mathbf{m}_{:\to \mathrm{i}}]_{\mathrm{k}})$
    Update replay buffer and policy $\hat{\pi}_i$
  end for
end while

We introduce four types of testing environments as shown in Fig. 2 in our paper, including Hallway [66], Level-Based Foraging (LBF) [45], Traffic Junction (TJ) [9], and two maps named 1o2r_vs_4r and 1o10b_vs_1r requiring communication from StarCraft Multi-Agent Challenge (SMAC) [66]. In this part, we will describe the details of these used environments.

Hallway. We design two instances of the Hallway environment, where agent can see nothing except its own location. In the first instance, we apply three hallways with lengths of $4, 5$, and 6, respectively. That means we let three agents a, b, c respectively initialized randomly at states a_1 to a_4, b_1 to b_5, and c_1 to c_6, and require them to arrive at state g

Algorithm C2 AME in the testing phase

Input: env, ablation size k, trained message-ablation policy $\hat{\pi}_i$ for each agent i

1: **for** $i = 1$ to N **do**
2: Receive a set of messages $\mathbf{m}_{:\to \mathrm{i}}$ with no more than $C < \frac{N}{2}$ malicious messages, and update local history τ_i
3: Discrete action space:
4: Take action $a \leftarrow \arg\max_{a\in\mathcal{A}} \sum_{[\mathbf{m}]_\mathrm{k}\in\mathcal{H}_\mathrm{k}(\mathbf{m})} \mathbb{I}[\hat{\pi}_i(\tau_i, [\mathbf{m}]_k) = a]$
5: Continuous Action Space:
6: Take action $a \leftarrow \mathrm{Median}\{\hat{\pi}_\mathrm{i}(\tau_\mathrm{i}, [\mathbf{m}]_\mathrm{k}) : [\mathbf{m}]_\mathrm{k} \in \mathcal{H}_\mathrm{k}(\mathbf{m})\}$
7: **end for**

simultaneously. In the second instance, 4 agents are distributed in four hallways with lengths of $3, 3, 4$, and 4. The action space is {*still*, *left*, *right*}. A reward of 1 will be given if all the agents arrive at the goal g simultaneously. However, if any agent does not reach the goal g at the same time, the game will stop immediately, and obtain 0 reward.

Level-Based Foraging (LBF). We use a variant version of the original environment, where only one agent can observe the map and others can see nothing. On this basis, we use two environment instances with different configurations, both of which are 8×8 grid world, 1 foods, and at least 3 agents are required to catch the food when they gather together next to the food concurrently. One of them contains 3 agents and they need to complete the task in 25 time-steps while the other contains 4 agents but only 15 time-steps are provided for them. The action set for each agent includes staying still and moving in one of four directions, only when at least 3 agents catch the food, they receive a constant reward $r = 1$.

Traffic Junction (TJ). The simulated traffic junction consists of several cars driving on the predefined road, and they need to avoid collisions and be as fast as possible. We use the *easy* version of the Traffic Junction environments as we mainly focus on the robustness of the algorithm. The *slow* version has an agent number limit to 5 and the max rate at which to add cars is 0.3 while the *fast* version has an agent number limit to 4 and the max rate at which to add cars is 0.4. In both of these two instances, the road dimension is 7, and the sight of the agent is limited to 0, which means each agent can only observe a 1×1 field of view around it. The action space is {*gas*, *brake*}, and each active car driving on the road needs to pay a time penalty $r_t = -0.01$ at every time-step. When a collision occurs, there will be a collision penalty $r_{\text{collision}} = -10$.

StarCraft Multi-Agent Challenge (SMAC). We use two maps named 1o2r_vs_4r and 1o10b_vs_1r in SMAC, which are introduced in NDQ [66]. In 1o2r_vs_4r, an Overseer finds 4 Reapers, and the ally units, 2 Roaches, need to reach enemies and kill them. Similarly, 1o10b_vs_1r is a map full of cliffs, where an Overseer detects a Roach, and the randomly spawned ally units, 10 Banelings, are required to reach and kill the enemy. The action set consists of moving in one of four directions, attacking one of the enemies, and staying still, and agents receive sharing rewards when some enemy units' hit-point drop or win the battle.

Appendix D Implemention Detail and Hyperparameters

We train our CroMAC agents based on PYMARL with its default network structure and hyperparameters setting, except that different environments have different RNN hidden sizes. We use Adam optimizer with learning rate 0.0005 and other default hyperparameters. For the state encoder and decoder, we use an MLP with one hidden layer and ReLU activation. For the message encoder, we use the same structure as the state encoder with shared parameters. All the prior distributions in the experiment are set to standard normal distribution, i.e., $\mathcal{N}(\mathbf{0}, \mathbf{1})$. The other hyperparameters of our proposed CroMAC for different benchmarks are summarized in Tab. D1.

Fast Gradient Sign Method (FGSM) [16] is a popular white-box method of generating adversarial examples, for MA-Dec-POMDP-Com, agent i receives multiple messages m_{ij}^t from teammate $j \in \{1, \cdots, i-1, i+1, \cdots, N\}$ at time t, we can compute perturbations for each m_{ij}^t with individual Q-network θ_i:

$$\eta_{ij}^t = \boldsymbol{\epsilon} \cdot \mathrm{sign}(\nabla_{m_{ij}^t} J(\theta_i, m_{ij}^t, y)),$$

where y is the originally selected action and the perturbed example is:

$$\hat{m}_{ij}^t = m_{ij}^t + \eta_{ij}^t.$$

Projected Gradient Descent (PGD) [36] can be regarded as an advanced version of FGSM where we implement it by adding perturbations within budget $\frac{\boldsymbol{\epsilon}}{3}$ to original message 3 times.

Table D1 Hyperparameters in experiments.

Hyperparameter	Hallway: 4x5x6 Hallway: 3x3x4x4	LBF: 3p-1f LBF: 4p-1f	TJ: slow TJ: fast	1o10b_vs_1r 1o2r_vs_4r
RNN Hidden Dim	16	32	32	64
Z Dim	16	32	32	64
VAE Hidden Dim	32	64	64	128
α_1	0.1	0.01	0.01	0.01
α_2	0.001	0.001	0.001	0.01
α_3	0.3	0.3	0.3	0.3 0.1
κ	5 10	5 10	10	10
C_MAX	0.1 0.2	0.3	0.3 0.6	0.3 0.2
Robust Start Time T_r	0.7M	0.8M	1.0M	1.0M
ϵ of FGSM (1)	0.3 \	0.02 \	0.0003 \	0.0055 \
ϵ of FGSM (2)	0.4 \	0.25 \	0.0004 \	0.0065 \
ϵ of FGSM	0.5	0.03 0.05	0.0005 0.001	0.0075 0.015
ϵ of FGSM (3)	0.6 \	0.35 \	0.0006 \	0.00875 \
ϵ of FGSM (4)	0.7 \	0.4 \	0.0007 \	0.01 \