Certifiably Robust Multi-Agent Reinforcement Learning against Adversarial Communication

Yanchao Sun [1] Ruijie Zheng [1] Parisa Hassanzadeh [2] Yongyuan Liang [3]
Soheil Feizi [1] Sumitra Ganesh [2] Furong Huang [1]

Abstract

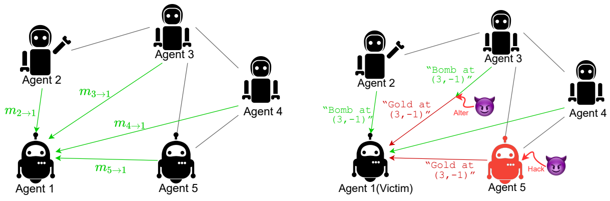
Communication is important in many multi-agent reinforcement learning (MARL) problems for agents to share information and make good decisions. However, when deploying trained communicative agents in a real-world application where noise and potential attackers exist, the safety of communication-based policies becomes a severe issue that is underexplored. Specifically, if communication messages are manipulated by malicious attackers, agents relying on untrustworthy communication may take unsafe actions that lead to catastrophic consequences. Therefore, it is crucial to ensure that agents will not be misled by corrupted communication, while still benefiting from benign communication. In this work, we consider an environment with N agents, where the attacker may arbitrarily change the communication from any $C < \frac{N-1}{2}$ agents to a victim agent. For this strong threat model, we propose a certifiable defense by constructing a message-ensemble policy that aggregates multiple randomly ablated message sets. Both theoretical analysis and experimental results verify that the message-ensemble policy can utilize benign communication while being certifiably robust to adversarial communication, regardless of the attacking algorithm.

1. Introduction

In a multi-agent system, especially in a cooperative game, communication usually plays an important role. By feeding communication messages as additional inputs to the policy network, each agent can obtain more information about the environment and other agents, and thus can learn a better policy (Foerster et al., 2016; Hausknecht, 2016).

However, communication can be a double-edged sword. A well-trained policy relying on communication can obtain high reward in a clean environment (e.g. a simulator), but it may get drastically misled by inaccurate information or even adversarial communication in a real-world application. Specifically, agents may receive misinformation due to hardware failures in a hostile environment; communication channels may be eavesdropped on by adversarial attackers, and messages may be altered maliciously; an attacker may also hack some agents and alter the messages they send to other agents (e.g. hacking IoT devices that usually lack sufficient protection (Naik & Maral, 2017)). Figure 1 shows an example of communication attacks, where the agents are trained with benign communication, but attackers may perturb the communication during test time. The attacker may lure a well-trained agent to a dangerous location through malicious message propagation and cause fatal damage.

Therefore, it is crucial to robustify MARL policies against adversarial communication, while still benefiting from benign communication. This is a challenging problem because **(1)** communication attacks can be stealthy and hard to identify (e.g. replace a word in a sentence as in Figure 1b); **(2)** the attacker's algorithm is unknown and can be adaptive to the victim's policy; **(3)** there can be more than one attacker (or an attacker can perturb more than one message at one step), such that they can collaborate to mislead a victim



(a) Training time **(b)** Test time (in deployment)

Figure 1. An example of test-time communication attacks in a communicative MARL system. (a) During training, agents are trained collaboratively in a safe environment, such as a simulator. (b) In deployment, agents execute pre-trained policies in the real world, where malicious attackers may modify benign (green) messages into adversarial (red) signals to mislead some victim agent(s).

Part of this work was done when the first author was an intern at JPMorgan AI Research. [1]University of Maryland, College Park, USA [2]JPMorgan AI Research, New York, USA [3]Sun Yat-sen University, China. Correspondence to: Yanchao Sun <ycs@umd.edu>.

ICML 2022 Workshop on Responsible Decision Making in Dynamic Environments, Baltimore, Maryland, USA, 2022. Copyright 2022 by the author(s).

agent. Some recent works (Blumenkamp & Prorok, 2020; Xue et al., 2021; Mitchell et al., 2020) take the first step to investigate adversarial communications in MARL and propose several defending methods, such as a learned message filter (Xue et al., 2021). However, these empirical defenses do not fully address the aforementioned challenges, and are not guaranteed to be robust, especially under adaptive attacks. More related work is discussed in Appendix B.

In this paper, we address all aforementioned challenges by proposing a certifiable defense, named **Ablated Message Ensemble (AME)**, that can guarantee the performance of agents when a fraction of communication messages are perturbed. The main idea of AME is to make decisions based on multiple different subsets of communication messages (i.e., ablated messages). Specifically, for a list of messages coming from different agents, we train a *message-ablation policy* that takes in a subset of messages and outputs a *base action*. Then, we construct an *message-ensemble policy* by aggregating multiple base actions coming from multiple ablated message subsets. For a discrete action space, the ensemble policy takes the majority of the multiple base actions obtained, while for a continuous action space, the ensemble policy takes the median of these base actions.

Our contributions can be summarized as below:
(1) We formulate the problem of adversarial attacks and defenses in communicative MARL (CMARL).
(2) We propose a novel defense, AME, that is certifiably robust against arbitrary perturbations of up to $C < \frac{N-1}{2}$ communications, where N is the number of agents.
(3) Experiment in several multi-agent environments shows that AME obtains significantly higher reward than baseline methods under both non-adaptive and adaptive attackers.

2. Problem Setup

Communicative Muti-agent Reinforcement Learning (CMARL). We consider a Decentralised Partially Observable Markov Decision Process (Dec-POMDP) (Oliehoek, 2012; Oliehoek & Amato, 2015; Das et al., 2020) which is a multi-agent generalization of the single-agent POMDP models. A Dec-POMDP can be modeled as a tuple $\langle \mathcal{D}, \mathcal{S}, \mathcal{A}_\mathcal{D}, \mathcal{O}_\mathcal{D}, O, P, R, \gamma \rangle$. $\mathcal{D} = \{1, \cdots, N\}$ is the set of N agents. $\mathcal{S}$ is the underlying state space. $\mathcal{A}_\mathcal{D} = \times_{i \in \mathcal{D}} \mathcal{A}_i$ is the *joint* action space. $\mathcal{O}_\mathcal{D} = \times_{i \in \mathcal{D}} \mathcal{O}_i$ is the *joint* observation space, with O being the observation emission function. P and R are the state transition function and the reward function, and γ is the discount factor.
Due to the partial observability, communication among agents is crucial for them to obtain high rewards. Consider a shared message space $\mathcal{M}$, where a message $m \in \mathcal{M}$ can be a scalar or a vector, e.g., signal of GPS coordinates. The communication policy of agent $i \in \mathcal{D}$, denoted as ξ_i, generates messages based on the agent's observation or interaction history. At every step t, agent i sends a communication message $m_{i \to j}^{(t)}$ to agent j for all $j \neq i$. (For notational simplicity, we will later omit $^{(t)}$ when there is no ambiguity.) We assume that agents are fully connected for communication, although our algorithm directly applies to partially connected communication graphs. That is, at each step, every agent sends out $N-1$ messages, and receives $N-1$ messages from others. This paper studies a general defense under any given communication protocol, so we do not restrict how the communication policy ξ is obtained (e.g., from a pre-defined communication protocol, or a learning algorithm (Foerster et al., 2016; Das et al., 2020)).
The goal of each agent $i \in \mathcal{D}$ is to maximize the discounted cumulative reward $\sum_{t=0}^{\infty} \gamma^t r^{(t)}$ by learning an acting policy π_i. When there exists communication, the policy input contains both its own interaction history, denoted by $\tau_i \in \Gamma_i$, and the communication messages $\mathbf{m}_{:\to i} := \{m_{j \to i} | 1 \leq j \leq N, j \neq i\}$. Similar to the communication policy ξ, we do not make any assumption on how the acting policy π is learned, as our defense mechanism introduced later can be plugged into any policy learning procedure.

Communication Attacks in Deployment of CMARL. During test time, agents execute well-trained policies. As shown in Figure 1b, the attacker may perturb communication messages to mislead a specific victim agent. Without loss of generality, suppose $i \in \mathcal{D}$ is the victim agent receiving $N-1$ communication messages from other agents. We consider the sparse attack setup where up to C messages can be arbitrarily perturbed at every step, among all $N-1$ messages. Here C is a reflection of the attacker's attacking power. The victim agent has no knowledge of which messages are adversarial. Note that this threat model is strong and can cover more attack scenarios than the commonly used ℓ_p attack. We illustrate the comparison between the above threat model and other attack models (ℓ_p attack, observation attack, action attack, etc.) in Appendix A.

We do not make any assumption on what attack algorithm the attacker uses, i.e., how a message is perturbed. In practice, an attacker may randomly change the communication signal, or learn a function to perturb the communication based on the current situation. The attacker can either be white-box or black-box, based on whether it knows the victim's policy and reward, as extensively studied in the field of adversarial supervised learning (Chakraborty et al., 2018).

3. Provably Robust Defense for CMARL

In this section, we present our defense algorithm, *Ablated Message Ensemble (AME)*, against test-time communication attacks in CMARL. We first make the following mild assumption for the attacking power.

Assumption 3.1 (Attacking Power)**.** An attacker can arbitrarily manipulate fewer than a half of the communication messages, i.e., $C < \frac{N-1}{2}$.

This is a realistic assumption, as it is less likely that an attacker can change the majority of communications among agents without being detected. Moreover, communications can be corrupted due to hardware failures, which usually affect a limited fraction of communications.

Our goal is to learn and execute a robust policy for any agent in the environment, so that the agent can perform well in both a non-adversarial environment and an adversarial environment. To ease the illustration, we focus on robustifying an arbitrary agent $i \in \mathcal{D}$, and the same defense is applicable to all other agents. We omit the agent subscript $_i$, and denote the agent's observation space, action space, and interaction history space as $\mathcal{O}$, $\mathcal{A}$, and Γ, respectively. Let $\mathbf{m} \in \mathcal{M}^{N-1}$ denote a set of $N-1$ messages received by the agent. Then, we can build an ablated message subset of $\mathbf{m}$ with k randomly selected messages, as defined below.

Definition 3.2 (***k***-Ablation Message Sample (k-Sample))**.** For a message set $\mathbf{m} \in \mathcal{M}^{N-1}$ and any integer $1 \leq k \leq N-1$, define a k-ablation message sample (or k-sample for short), $[\mathbf{m}]_k \in \mathcal{M}^k$, as a set of k randomly sampled messages from $\mathbf{m}$. Let $\mathcal{H}_k(\mathbf{m})$ be the collection of all unique k-samples of $\mathbf{m}$, and thus $|\mathcal{H}_k(\mathbf{m})| = \binom{N-1}{k}$.

We propose Ablated Message Ensemble (AME), a generic defense framework that can be fused with any policy learning algorithm. The **main idea** of AME is to make decisions based on the *consensus* of the benign messages. We train a base policy that makes decisions with one k-sample at each step. During test time when communication messages may be perturbed, the agent collects multiple k-samples at every step, and applies the trained base policy to each k-sample to get multiple resulting base actions. Then, the agent selects the action that reflects the majority opinion. By carefully designed ablation and ensemble strategies, we can ensure that the majority of these base actions is dominated by benign messages. The procedures of the training phase and the testing/defending phase of AME are illustrated below, and detailed by Algorithm 1 and Algorithm 2 in Appendix C.

Training Phase with Message-Ablation Policy $\hat{\pi}$ (Algorithm 1). During training, the agent learns a *message-ablation policy* $\hat{\pi} : \Gamma \times \mathcal{M}^k \to \mathcal{A}$ which maps its own interaction history τ and a random k-sample $[\mathbf{m}]_k \sim \mathrm{Uniform}(\mathcal{H}_k(\mathbf{m}))$ to an action, where $\mathrm{Uniform}(\mathcal{H}_k(\mathbf{m}))$ is the uniform distribution over $\mathcal{H}_k(\mathbf{m})$. Here k is a user-specified hyperparameter. A larger k can improve the natural performance, but a smaller k can improve the robustness. We analyze the selection of k and the corresponding theoretical guarantees in Appendix D.3. The training objective is to maximize the cumulative reward of $\hat{\pi}$ based on randomly sampled k-samples in a non-adversarial environment. Any policy optimization algorithm can be used for training.

Defending Phase with Message-Ensemble Policy $\widetilde{\pi}$ (Algorithm 2). The main idea of our test-time defense is to collect all possible k-samples from $\mathcal{H}_k(\mathbf{m})$, and select an action suggested by the majority of those k-samples, based on a well-trained message-ablation policy $\hat{\pi}$. Specifically, we construct a *message-ensemble policy* $\widetilde{\pi} : \Gamma \times \mathcal{M}^{N-1} \to \mathcal{A}$ that outputs an action by aggregating the base actions produced by $\hat{\pi}$ on multiple k-samples (Line 5 in Algorithm 2). The construction of the message-ensemble policy depends on whether the agent's action space $\mathcal{A}$ is discrete or continuous, which is given below by Definition 3.3 below.

Definition 3.3 (Message-Ensemble Policy)**.** For a message-ablation policy $\hat{\pi}$ with observation history τ and received message set $\mathbf{m}$, define the message-ensemble policy $\widetilde{\pi}$ as

$$\widetilde{\pi}(\tau, \mathbf{m}) := \mathrm{argmax}_{a \in \mathcal{A}} \textstyle\sum_{[\mathbf{m}]_k \in \mathcal{H}_k(\mathbf{m})} \mathbb{1}[\hat{\pi}(\tau, [\mathbf{m}]_k) = a], \tag{1}$$

for a discrete action space $\mathcal{A}$, and

$$\widetilde{\pi}(\tau, \mathbf{m}) = \mathsf{Median}\{\hat{\pi}(\tau, [\mathbf{m}]_k) : [\mathbf{m}]_k \in \mathcal{H}_k(\mathbf{m})\}, \tag{2}$$

for a continuous action space $\mathcal{A}$, where the function Median returns the element-wise median value of a set of vectors.

Therefore, the message-ensemble policy $\widetilde{\pi}$ takes the action suggested by the consensus of all k-samples (majority vote in a discrete action space, and median action for a continuous action space). A similar randomized ablation idea is used by Levine & Feizi (2020) to defend against ℓ_0 attacks in image classification. However, their high-probability guarantee for a single-step decision is not suitable for sequential decision-making problems, as the guaranteed probability decreases when it propagates over timesteps. Moreover, their algorithm does not work when the model has a continuous output space. In contrast, AME has robustness guarantees for both the immediate action and the long-term reward, and for both discrete and continuous actions.

Action Certificates of AME. We can prove that under mild conditions, the action selected by $\widetilde{\pi}$ is guaranteed to be within the range of "benign actions" that are suggested by purely-benign k-samples. Therefore, when an agent executes message-ensemble policy, its actions can be certified to be relatively safe (dominated by benign information instead of misinformation). Full analysis is in Appendix D.

Reward Certificates of AME. With the above action certificate, $\widetilde{\pi}$ under adversarial communications works similarly as the message-ablation policy $\hat{\pi}$ under all-benign communications. Therefore, we can further provide performance certificates of AME for the long-term reward. Appendix D provides detailed analysis and shows that the cumulative reward of $\widetilde{\pi}$ under adversarial communications is comparable with the natural reward of the message-ablation policy $\hat{\pi}$ under benign communications.

Extensions of AME. (1) Scale Up. The message-ensemble policy aggregates all $\binom{N-1}{k}$ k-samples from $\mathcal{H}_k(\mathbf{m})$, which is inexpensive when N is small. But if N is large, we can select only a certain number of samples from $\mathcal{H}_k(\mathbf{m})$, and a theoretical guarantee can still be derived, as shown in

Appendix D.4. **(2) Attack Detection.** The idea of AME can also be extended to detect attackers, by measuring the difference between the action suggested by any specific communication message and the ensemble policy's action, as detailed and empirically verified in Appendix G.

4. Empirical Study

In this section, we verify the robustness of our AME in multiple different CMARL environments against various communication attack algorithms.

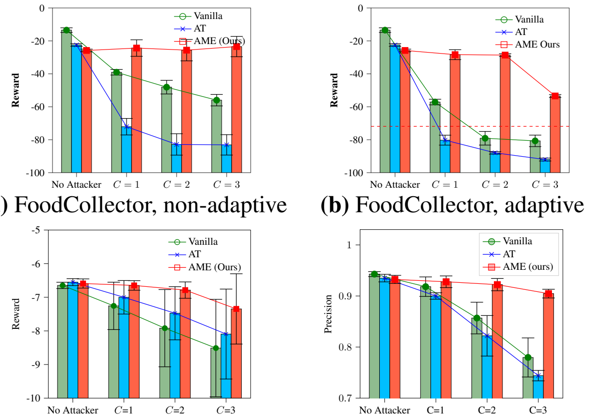
Environments. To evaluate the effectiveness of AME, we consider the following environments.

- *FoodCollector*: a 2D particle environment where $N = 9$ agents with different colors search for foods with the same colors. Agents share their local observations with each other to expedite food collecting.
- *InventoryManager*: a simulator of a real-world inventory management problem, where $N = 10$ distributor agents manage their inventory of multiple products to match an unknown product demand distribution. Agents can share observed demand requests to improve inventory management.
- *MARL-MNIST*: an MARL image classification problem (Mousavi et al., 2019) in the MNIST dataset. Each of $N = 9$ agents can observe a patch of an image at every step. Agents communicate with encoded observations (learned communication) to reach a classification decision.

Detailed description of each environment is in Appendix F.

Evaluation Metrics. To evaluate the effectiveness of our defense strategy, we test the performance of the trained policies under no attack and under various types of attacks with different C's (for simplicity, we refer to C as the *number of adversaries*). (1) *Non-adaptive attack* that perturbs messages based on heuristics (e.g. random message, permute dimensions, etc). (2) *Learned adaptive attack* that learns the strongest/worst adversarial communication with an RL algorithm to minimize the victim's reward (a white-box attacker which knows the victim's reward). As shown in prior work (Zhang et al., 2020), this RL attack formulation is a theoretically optimal attack (which minimizes the victim's reward). Therefore, we can regard this learned attack as a worst-case attack for the victim agents. The attack implementation is elaborated in Appendix F.

Implementation and Baselines. As discussed in Section 3, our AME is designed to defend against communication attacks, and can be fused with any policy learning algorithm. In FoodCollector and InventoryManager, we use PPO (Schulman et al., 2017) with parameter sharing (Terry et al., 2020) as the base policy learning algorithm as it achieves relatively high performance. In MARL-MNIST, we use the LSTM-based policy proposed by Mousavi et al. (2019). But our AME can be applied to other MARL algorithms, such as QMIX (Rashid et al., 2018) that is evaluated in Appendix F.1.3. Then, with the same policy learning method, we compare our AME with two defense baselines: (1) **(Vanilla)**: vanilla training without defense, which learns a policy based on all benign messages. (2) **(AT)** adversarial training as in Zhang et al. (2021), which alternately trains an adaptive RL attacker and an agent. During training and defending, we set the ablation size AME as $k = 2$, the largest solution to Equation (7) for $C = 2$ when $N = 9$ or $N = 10$. For AT, we train the agent against $C = 2$ learned adversaries. More implementation details and hyperparameter settings are provided in Appendix F.



(a) FoodCollector, non-adaptive **(b)** FoodCollector, adaptive

(c) InventoryManager, adaptive **(d)** MARL-MNIST, adaptive

Figure 2. Rewards of our AME and baselines under no attacker and varying numbers of adversaries for non-adaptive and adaptive attacks. Results are averaged over 5 random seeds.

Experiment Results. Figure 2 shows part of the empirical results, while the full results are in Appendix F. We can see that the rewards of Vanilla and AT drastically drop under attacks. Although AT is usually effective for ℓ_p attacks (Zhang et al., 2021), we find that AT does not achieve better robustness than Vanilla, since it can not adapt to arbitrary perturbations to several messages. In contrast, *AME can utilize benign communication well while being robust to adversarial communication.* We use $k = 2$ for our AME, which in theory provides performance guarantees against up to $C = 2$ adversaries for $N = 9$ and $N = 10$. We can see that the reward of AME under $C = 1$ or $C = 2$ is similar to its reward under no attack, matching our theoretical analysis. Even under 3 adversaries where the theoretical guarantees no longer hold, AME still obtains superior performance compared to Vanilla and AT. Therefore, *AME makes agents robust under varying numbers of adversaries.* Detailed hyperparameter tests, and empirical study on variants of AME are provided in Appendix F.

5. Conclusion

This paper formulates communication attacks in MARL problems, and proposes a defense framework AME, which is certifiably robust against multiple arbitrarily perturbed adversarial communication messages, based on randomized ablation and aggregation of messages.

Acknowledgements

This work is supported by JPMorgan Chase & Co., National Science Foundation NSF-IIS-FAI Award 2147276, Office of Naval Research, Defense Advanced Research Projects Agency Guaranteeing AI Robustness against Deception (GARD), and Adobe, Capital One and JP Morgan faculty fellowships.

Disclaimer This paper was prepared for informational purposes in part by the Artificial Intelligence Research group of JPMorgan Chase & Coȧnd its affiliates ("JP Morgan"), and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

Achiam, J. Spinning Up in Deep Reinforcement Learning. 2018.

Berrien, S. Marl classification. https://github.com/Ipsedo/MARLClassification, 2019.

Blumenkamp, J. and Prorok, A. The emergence of adversarial communication in multi-agent reinforcement learning, 2020.

Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., and Mukhopadhyay, D. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.

Chou, P.-W., Maturana, D., and Scherer, S. Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 834–843. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/chou17a.html.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

Das, A., Gervet, T., Romoff, J., Batra, D., Parikh, D., Rabbat, M., and Pineau, J. Tarmac: Targeted multi-agent communication, 2020.

Fischer, M., Mirman, M., Stalder, S., and Vechev, M. Online robustness training for deep reinforcement learning. *arXiv preprint arXiv:1911.00887*, 2019.

Foerster, J. N., Assael, Y. M., De Freitas, N., and Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. *arXiv preprint arXiv:1605.06676*, 2016.

Gleave, A., Dennis, M., Wild, C., Kant, N., Levine, S., and Russell, S. Adversarial policies: Attacking deep reinforcement learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HJgEMpVFwB.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T., and Kohli, P. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

Hausknecht, M. J. *Cooperation and communication in multiagent deep reinforcement learning*. PhD thesis, 2016.

Huang, S., Papernot, N., Goodfellow, I., Duan, Y., and Abbeel, P. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.

Kumar, A., Levine, A., and Feizi, S. Policy smoothing for provably robust reinforcement learning. *arXiv preprint arXiv:2106.11420*, 2021.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Levine, A. and Feizi, S. Robustness certificates for sparse adversarial attacks by randomized ablation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):4585–4593, Apr. 2020. doi: 10.1609/aaai.v34i04.5888. URL https://ojs.aaai.org/index.php/AAAI/article/view/5888.

Liu, Y.-C., Tian, J., Ma, C.-Y., Glaser, N., Kuo, C.-W., and Kira, Z. Who2com: Collaborative perception via learnable handshake communication. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6876–6883. IEEE, 2020.

Lütjens, B., Everett, M., and How, J. P. Certified adversarial robustness for deep reinforcement learning. In *Conference on Robot Learning*, pp. 1328–1337. PMLR, 2020.

Mallik, A. Man-in-the-middle-attack: Understanding in simple words. *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, 2(2):109–134, 2019.

Mitchell, R., Blumenkamp, J., and Prorok, A. Gaussian process based message filtering for robust multi-agent cooperation in the presence of adversarial communication, 2020.

Mousavi, H. K., Nazari, M., Takáč, M., and Motee, N. Multi-agent image classification via reinforcement learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5020–5027. IEEE, 2019.

Naik, S. and Maral, V. Cyber security — iot. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pp. 764–767, 2017. doi: 10.1109/RTEICT.2017.8256700.

Oikarinen, T., Zhang, W., Megretski, A., Daniel, L., and Weng, T.-W. Robust deep reinforcement learning through adversarial loss. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=eaAM_bdW0Q.

Oliehoek, F. A. Decentralized pomdps. In *Reinforcement Learning*, pp. 471–503. Springer, 2012.

Oliehoek, F. A. and Amato, C. A concise introduction to decentralized pomdps, 2015.

Phan, T., Belzner, L., Gabor, T., Sedlmeier, A., Ritz, F., and Linnhoff-Popien, C. Resilient multi-agent reinforcement learning with adversarial value decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11308–11316, 2021.

Pinto, L., Davidson, J., Sukthankar, R., and Gupta, A. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, pp. 2817–2826. PMLR, 2017.

Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018.

Rashid, T., Samvelyan, M., de Witt, C. S., Farquhar, G., Foerster, J. N., and Whiteson, S. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning.*, 2018.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shen, Q., Li, Y., Jiang, H., Wang, Z., and Zhao, T. Deep reinforcement learning with robust and smooth policy. In *International Conference on Machine Learning*, pp. 8707–8718. PMLR, 2020.

Sukhbaatar, S., Fergus, R., et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29:2244–2252, 2016.

Sun, Y., Zheng, R., Liang, Y., and Huang, F. Who is the strongest enemy? towards optimal and efficient evasion attacks in deep rl. *arXiv preprint arXiv:2106.05087*, 2021.

Terry, J. K., Grammel, N., Son, S., and Black, B. Parameter sharing for heterogeneous agents in multi-agent reinforcement learning. *CoRR*, abs/2005.13625, 2020. URL https://arxiv.org/abs/2005.13625.

Tessler, C., Efroni, Y., and Mannor, S. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*, pp. 6215–6224. PMLR, 2019.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Tu, J., Wang, T., Wang, J., Manivasagam, S., Ren, M., and Urtasun, R. Adversarial attacks on multi-agent communication. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7768–7777, October 2021.

Wu, F., Li, L., Huang, Z., Vorobeychik, Y., Zhao, D., and Li, B. Crop: Certifying robust policies for reinforcement learning through functional smoothing. *arXiv preprint arXiv:2106.09292*, 2021.

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

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pp. 7472–7482. PMLR, 2019.

Zhang, H., Chen, H., Xiao, C., Li, B., Liu, M., Boning, D., and Hsieh, C.-J. Robust deep reinforcement learning against adversarial perturbations on state observations. *arXiv preprint arXiv:2003.08938*, 2020.

Zhang, H., Chen, H., Boning, D., and Hsieh, C.-J. Robust reinforcement learning on state observations with learned optimal adversary. *arXiv preprint arXiv:2101.08452*, 2021.

Appendix: Certifiably Robust Multi-Agent Reinforcement Learning against Adversarial Communication

A. Discussion on Threat Model

This section compares the ours threat model established in Section 2 and other closely related threat models.

Comparison with ℓ_p Threat Models Many existing works on adversarial attack and defense (Goodfellow et al., 2014; Huang et al., 2017; Zhang et al., 2020; Tu et al., 2021) assume that the perturbation is small in ℓ_p norm. However, many realistic and stealthy attacks can not be covered by the ℓ_p threat model. For example, the attacker may replace a word in a sentence as in Figure 1b, add a patch to an image, or shift the signal by some bits. In these cases, the ℓ_p distance between the clean value and the perturbed value is large, such that ℓ_p defenses do not work. In contrast, these attacks are covered by our threat model which allows arbitrary perturbations to C messages. Therefore, our setting can work for broader applications.

Comparison between Communication Attacks and Other Attacks in MARL Adversarial attacks and defenses in RL systems have recently attracted more and more attention, and are considered in many different scenarios. The majority of related work focuses on *directly attacking a victim* by perturbing its observations (Huang et al., 2017; Gleave et al., 2020; Oikarinen et al., 2021; Sun et al., 2021) or actions (Tessler et al., 2019; Pinto et al., 2017). However, an attacker may not have direct access to the specific victim's observation or action. In this case, *indirect attacks via other agents* can be an alternative. For example, Gleave et al. (2020) propose to attack the victim by changing the other agent's actions. Therefore, even if the victim agent has well-protected sensors, the attacker can still influence it by manipulating other under-protected agents. But the intermediary agent whose actions are altered will obtain sub-optimal reward, which makes the attack noticeable and less stealthy. In contrast, if an attacker alters the communication messages sent from the other agents (e.g., by man-in-the-middle attacks (Mallik, 2019)), the behaviors of other agents are not changed, and thus it is relatively hard to find who has sent adversarial messages and which messages are not trustworthy.
It is also worth pointing out that, since an acting policy π takes in both its own observation and the communication messages, communication can be regarded as a subset of policy inputs.

Relation between Communication Attack and ℓ_0 Observation Attack Our communication threat model is analogous to a constrained ℓ_0 attack on policy inputs (Levine & Feizi, 2020). When agent i is attacked, the input space of its acting policy π_i is $\mathcal{X}_i := \Gamma_i \times \mathcal{M}^{N-1}$. Therefore, when C communication messages are corrupted, the original input x_i gets perturbed to $\tilde{x}_i$. Let d be the dimension of a communication message, then x_i and $\tilde{x}_i$ differ by up to dC dimensions, which is similar to an ℓ_0 attack constrained in certain dimensions.

B. Related Work

Adversarial Robustness of RL Agents As RL techniques have been applied to more and more applications, the robustness of agents under environment noise or adversarial attacks becomes an emerging research topic. Section A introduces several adversarial attacks on single-agent and multi-agent problems. To improve the robustness of agents, adversarial training (i.e., introducing adversarial agents to the system during training (Pinto et al., 2017; Phan et al., 2021; Zhang et al., 2021; Sun et al., 2021)) and network regularization (Zhang et al., 2020; Shen et al., 2020; Oikarinen et al., 2021) are empirically shown to be effective under ℓ_p attacks, although such robustness is not theoretically guaranteed. Certifying the robustness of a network is an important research problem (Raghunathan et al., 2018; Cohen et al., 2019; Gowal et al., 2018). In an effort to certify RL agents' robustness, some approaches (Lütjens et al., 2020; Zhang et al., 2020; Oikarinen et al., 2021; Fischer et al., 2019) apply network certification tools to bound the Q networks and improve the worst-case value of the policy. Kumar et al. (2021) and Wu et al. (2021) follow the idea of randomized smoothing (Cohen et al., 2019) and smooth out the policy by adding Gaussian noise to the input.

Communication in MARL Communication is crucial in solving collaborative MARL problems. There are many existing studies learning communication protocols across multiple agents. Foerster et al. (2016) are the first to learn differentiable communication protocols that is end-to-end trainable across agents. Another work by Sukhbaatar et al. (2016) proposes an efficient permutation-invariant centralized learning algorithm which learns a large feed-forward neural network to map inputs of all agents to their actions. It is also important to communicate selectively, since some communication may be less informative or unnecessarily expensive. To tackle this challenge, Das et al. (2020) propose an attention mechanism for agents to adaptively select which other agents to send messages to. Liu et al. (2020) introduce a handshaking procedure so that the agents communicate only when needed. Our work assumes a pre-trained communication protocol and does not consider the problem of learning communication mechanisms.

Adversarial Attacks and Defenses in CMARL Recently, the problem of adversarial communication in MARL has attracted increasing attention. Blumenkamp & Prorok (2020) show that in a cooperative game, communication policies of some self-interest agents can hurt other agents' performance. To achieve robust communication, (Mitchell et al., 2020) adopt a Gaussian Process-based probabilistic model to compute the posterior probabilities that whether each partner is truthful. Tu et al. (2021) investigate the vulnerability of multi-agent autonomous systems against communication attacks, with a focus on vision tasks. Xue et al. (2021) propose an algorithm to defend against one adversarial communication message by an anomaly detector and a message reconstructor, which are trained with groundtruth labels and messages.

To the best of our knowledge, our AME is the first certifiable defense in MARL against communication attacks. Moreover, we consider a strong threat model where up to half of the communication messages can be arbitrarily corrupted, capturing many realistic types of attacks.

C. Additional Algorithm Details

Algorithm 1 and Algorithm 2 demonstrate the procedures of AME in training and defending phases, respectively.

Algorithm 1 Training Phase of AME

```
Input: ablation size $k$
Initialize $\hat{\pi}_i$ for every agent $i \in [N]$.
repeat
    for $i = 1$ to $N$ do
        Receive a list of messages $\mathbf{m}_{:\to i}$, get local observation $o_i$ and update interaction history $\tau_i$
        Randomly sample $[\mathbf{m}_{:\to i}]_k \sim \mathrm{Uniform}(\mathcal{H}_k(\mathbf{m}_{:\to i}))$
        Take action based on $\tau_i$ and the k-sample $[\mathbf{m}_{:\to i}]_k$, i.e., $a_i \leftarrow \hat{\pi}_i(\tau_i, [\mathbf{m}_{:\to i}]_k)$
        Update the replay buffer and policy $\hat{\pi}_i$
    end for
until end of training
Output: message-ablation policy $\hat{\pi}_i, \forall i \in [N]$
```

Algorithm 2 Defending Phase of AME

```
Input: ablation size $k$; trained message-ablation policy $\hat{\pi}_i, \forall i \in [N]$,
repeat
    for $i = 1$ to $N$ do
        Receive a list of messages $\mathbf{m}_{:\to i}$ with at most $C$ malicious messages, get local observation $o_i$ and update interaction history $\tau_i$
        Take $\widetilde{a}_i \leftarrow \widetilde{\pi}_i(\tau_i, \mathbf{m}_{:\to i})$, where $\widetilde{\pi}_i$ is the message-ensemble policy defined with $\hat{\pi}$ by Equation (1) for discrete $\mathcal{A}_i$, and Equation (2) for continuous $\mathcal{A}_i$
    end for
until end of test
```

D. Theoretical Analysis for AME

In this section, we provide theoretical guarantees for the robustness of AME. During test time, at any step, let τ be the interaction history, $\mathbf{m}_{\mathrm{benign}}$ be the unperturbed benign message set, and $\mathbf{m}_{\mathrm{adv}}$ be the perturbed message set. Note that $\mathbf{m}_{\mathrm{benign}}$ and $\mathbf{m}_{\mathrm{adv}}$ both have $N-1$ messages while they differ by up to C messages. With the above notation, we define a set of actions rendered by purely benign k-samples in Definition D.1. As the agent using a well-trained message-ablation policy is likely to take these actions in a non-adversarial environment, they can be regarded as "good" actions to take.

Definition D.1 (Benign Action Set $\mathcal{A}_{\mathrm{benign}}$)**.** For the execution of the message-ablation policy $\hat{\pi}$ at any step, define $\mathcal{A}_{\mathrm{benign}} \subseteq \mathcal{A}$ as a set of actions that $\hat{\pi}$ may select under benign k-samples.

$$\mathcal{A}_{\mathrm{benign}} := \cup_{[\mathbf{m}_{\mathrm{benign}}]_k \in \mathcal{H}_k(\mathbf{m}_{\mathrm{benign}})} \left\{ \hat{\pi}(\tau, [\mathbf{m}_{\mathrm{benign}}]_k) \right\}. \tag{3}$$

Next, in Section D.1 and D.2, we provide the theoretical analysis and guarantees for discrete action spaces and continuous action spaces, respectively.

D.1. Robustness Certificates of AME in Discrete Action Space

For a discrete action space, the message-ablation policy takes the action with the most votes from all k-samples as suggested by Equation (1). To ensure that this action stands for the consensus of benign messages, the following condition is needed.

Condition D.2 (Confident Consensus)**.** Denote $u_{\max} := \max_{a \in \mathcal{A}} \sum_{[\mathbf{m}]_k \in \mathcal{H}_k(\mathbf{m})} \mathbb{1}[\hat{\pi}(\tau, [\mathbf{m}]_k) = a]$ as the highest number of votes among all actions, and $u_{\max}$ satisfies

$$u_{\max} > \binom{N-1}{k} - \binom{N-1-C}{k} =: u_{\text{adv}}, \tag{4}$$

where u_{adv} is the number of votes that adversarial messages may affect (number of k-samples that contain at least one adversarial message).

Remarks. (1) This condition ensures the consensus has more votes than the votes that adversarial messages are involved in. Therefore, when $\widetilde{\pi}$ takes an action, there must exist some purely-benign k-samples voting for this action. (2) This condition is easy to satisfy when $C \ll N$ as $\binom{N-1}{k} \approx \binom{N-1-C}{k}$. (3) This condition can be easily checked at every step of execution. (4) We analyze the relation between this condition and the selection of k in Appendix D.3.

Condition D.2 considers the worst-case scenario when the adversarial messages collaborate to vote for a malicious action and outweigh benign messages in all k-samples. However, such a worst-case attack is uncommon in practice as attackers are usually not omniscient. Therefore, Condition D.2 is sufficient but not necessary for the robustness of $\widetilde{\pi}$ during execution. In real-world problems, our algorithm achieves strong robustness without requiring this condition, as verified in experiments.

We now provide both action and reward certificates for the ensemble policy $\widetilde{\pi}$ in the discrete-action case.

Action Certificate for Discrete Action Space The following theorem suggests that the ensemble policy $\widetilde{\pi}$ always takes benign actions under the above conditions no matter whether attacks exist.

Theorem D.3 (Action Certificate for Discrete Action Space)**.** *Under Condition D.2, the ensemble policy $\widetilde{\pi}$ in Equation* (1) *produces benign actions under $\mathbf{m}_{\text{adv}}$, i.e.,*

$$\widetilde{a} = \widetilde{\pi}(\tau, \mathbf{m}_{\text{adv}}) \in \mathcal{A}_{\text{benign}}. \tag{5}$$

Under Condition D.5 and Condition D.2 which are easy to check, Theorem D.3 certifies that $\widetilde{\pi}$ ignores the malicious messages in $\mathbf{m}_{\text{adv}}$ and executes a benign action that is suggested by some benign message combinations, even when the malicious messages are not identified. Then, we can further derive a reward certificate as introduced below.

Reward Certificate for Discrete Action Space Theorem D.3 justifies that under sufficient majority votes, the message-ensemble policy $\widetilde{\pi}$ ignores the malicious messages in $\mathbf{m}_{\text{adv}}$ and executes a benign action that is suggested by some benign message combinations, even when the malicious messages are not identified. It is important to note that, when the message-ensemble policy $\widetilde{\pi}$ selects an action $\widetilde{a}$, there must exist at least one purely benign k-sample that let the message-ablation policy $\hat{\pi}$ produce $\widetilde{a}$. Therefore, as long as $\hat{\pi}$ can obtain high reward with randomly selected benign k-samples, $\widetilde{\pi}$ can also obtain high reward with ablated adversarial communication due to its design.

Specifically, we consider a specific agent with message-ablation policy $\hat{\pi}$ and message-ensemble policy $\widetilde{\pi}$ (suppose other agents are executing fixed policies). Let $\nu : \mathcal{M}^{N-1} \to \mathcal{M}^{N-1}$ be an attack algorithm that perturbs at most C messages in a message set. Let $\zeta \sim Z(P, \hat{\pi})$ be a trajectory of policy $\hat{\pi}$ under no attack, i.e., $\zeta = (o^{(0)}, \mathbf{m}^{(0)}, a^{(0)}, r^{(0)}, o^{(1)}, \mathbf{m}^{(1)}, a^{(0)}, r^{(0)}, \cdots)$. (Recall that a message-ablation policy $\hat{\pi}$ takes in a random size-k subset of $\mathbf{m}^{(t)}$ and outputs action $a^{(t)}$.) When there exists attack with ν, let $\zeta_\nu \sim Z(P, \widetilde{\pi}; \nu)$ be a trajectory of policy $\widetilde{\pi}$ under communication attacks, i.e., $\zeta_\nu = (o^{(0)}, \nu(\mathbf{m}^{(0)}), a^{(0)}, r^{(0)}, o^{(1)}, \nu(\mathbf{m}^{(1)}), a^{(0)}, r^{(0)}, \cdots)$. For any trajectory ζ, let $r(\zeta)$ be the discounted cumulative reward of this trajectory.

With the above notations, we propose the following reward certificate.

Corollary D.4 (Reward Certificate for Discrete Action Space)**.** *When Condition D.2 holds at every step of execution, the cumulative reward of ensemble policy $\widetilde{\pi}$ defined in Equation* (1) *under adversarial communication is no lower than the lowest cumulative reward that the ablation policy $\hat{\pi}$ can obtain with randomly selected k-samples under no attacks, i.e.,*

$$\min_{\zeta_\nu \sim Z(P, \widetilde{\pi}; \nu)} r(\zeta_\nu) \geq \min_{\zeta \sim Z(P, \hat{\pi})} r(\zeta), \tag{6}$$

for any attacker ν satisfying Assumption 3.1 ($C < \frac{N-1}{2}$).

Remarks. (1) The certificate holds for any attack algorithm ν with $C < \frac{N-1}{2}$. (2) The message-ablation policy $\hat{\pi}$ has extra randomness from the sampling of k-samples. That is, at every step, $\hat{\pi}$ takes a uniformly randomly selected k-sample from $\mathcal{H}_k(\mathbf{m})$. Therefore, the $\min_\zeta$ in the RHS considers the worst-case message sampling in the clean environment without attacks. Since $\widetilde{\pi}$ always takes actions selected by some purely-benign message combinations, the trajectory generated by $\widetilde{\pi}$ can also be produced by the message-ablation policy. (3) Note that the RHS ($\min_{\zeta \sim Z(P,\hat{\pi})} r(\zeta)$) can be approximately estimated by executing $\hat{\pi}$ during training time, so that the test-time performance of $\widetilde{\pi}$ is guaranteed to be no lower than this value, even if there are up to C corrupted messages at every step.

Appendix E provides detailed proofs for the above theoretical results.

D.2. Robustness Certificates of AME in Continuous Action Space

For a continuous action space, the message-ensemble policy takes the element-wise median of all base actions. To ensure that this action taken is relatively safe, we characterize a condition for the ablation size k as below.

Condition D.5 (Dominating Benign Sample)**.** The ablation size k of AME satisfies

$$\binom{N-1-C}{k} > \frac{1}{2}\binom{N-1}{k}. \tag{7}$$

Remarks. (1) For the message set $\mathbf{m}_{\text{adv}}$ that has up to C adversarial messages, Equation (7) implies that among all k-samples from $\mathbf{m}_{\text{adv}}$, there are more purely benign k-samples than k-samples that contain adversarial messages. (2) Under Assumption 3.1, this condition *always has solutions* for k as $C < \frac{N-1}{2}$, and $k = 1$ is always a feasible solution.

Next, we provide both action certificate and reward certificate for the message-ensemble policy $\widetilde{\pi}$ in a continuous action space.

Action Certificate As in the discrete action case, the continuous ensemble policy will output an action $\widetilde{a}$ that follows the consensus of benign messages. But in a continuous action space, it is hard to ensure that $\widetilde{a}$ is exactly in $\mathcal{A}_{\text{benign}}$. Instead, $\widetilde{a}$ is guaranteed to be within the range of $\mathcal{A}_{\text{benign}}$, as detailed in the following theorem.

Theorem D.6 (Action Certificate for Continuous Action Space)**.** *Under Condition D.5, the action $\widetilde{a} = \widetilde{\pi}(\tau, \mathbf{m}_{\text{adv}})$ generated by the ensemble policy $\widetilde{\pi}$ defined in Equation (2) satisfies*

$$\widetilde{a} \in \mathsf{Range}(\mathcal{A}_{\text{benign}}) := \{a : \forall 1 \le l \le L, \exists \underline{a}, \overline{a} \in \mathcal{A}_{\text{benign}} \textit{ s.t } \underline{a}_l \le a_l \le \overline{a}_l\}. \tag{8}$$

In other words, if the message-ensemble policy takes an action $\widetilde{a}$, then at each dimension l, there must exist a_1 and a_2 suggested by some purely benign k-samples such that the l-th dimension of $\widetilde{a}$ is lower and upper bounded by the l-th dimension of a_1 and a_2, respectively. Note that Theorem D.6 certifies that the selected action is in a set of actions that are close to benign actions $\mathsf{Range}(\mathcal{A}_{\text{benign}})$, but does not make any assumption on this set. Next we interpret this result in details.

How to Understand $\mathsf{Range}(\mathcal{A}_{\text{benign}})$? Theoretically, $\mathsf{Range}(\mathcal{A}_{\text{benign}})$ is a set of actions that are coordinate-wise bounded by base actions resulted from purely benign k-samples. In many practical problems, it is reasonable to assume that actions in $\mathsf{Range}(\mathcal{A}_{\text{benign}})$ are relatively safe, especially when benign actions in $\mathcal{A}_{\text{benign}}$ are concentrated. The following examples illustrate some scenarios where actions in $\mathsf{Range}(\mathcal{A}_{\text{benign}})$ are relatively good.

1. If the action denotes the price a seller sells its product in a market, and the communication messages are the transaction signals in an information pool, then $\mathsf{Range}(\mathcal{A}_{\text{benign}})$ is a price range that is determined based on purely benign messages. Therefore, the seller will set a reasonable price without being influenced by a malicious message.

2. If the action denotes the driving speed, and benign message combinations have suggested driving at 40 mph or driving at 50 mph, then driving at 45 mph is also safe.

3. If the action is a vector denoting movements of all joints of a robot (as in many MuJoCo tasks), and two slightly different joint movements are suggested by two benign message combinations, then an action that does not exceed the range of the two benign movements at every joint is likely to be safe as well.

The above examples show by intuition why the message-ensemble policy can be regarded as a relatively robust policy. However, in extreme cases where there exists "caveat" in $\mathsf{Range}(\mathcal{A}_{\text{benign}})$, taking an action in this set may also be unsafe. To quantify the influence of $\mathsf{Range}(\mathcal{A}_{\text{benign}})$ on the long-term reward, we next analyze the cumulative reward of the message-ensemble policy in the continuous-action case.

How Does $\mathsf{Range}(\mathcal{A}_{\text{benign}})$ Lead to A Reward Certificate? Different from the discrete-action case, the message-ensemble policy $\widetilde{\pi}$ in a continuous action space may take actions not in $\mathcal{A}_{\text{benign}}$ such that it generates trajectories not seen by the message-ablation policy $\hat{\pi}$. However, since the action of $\widetilde{\pi}$ is guaranteed to stay in $\mathsf{Range}(\mathcal{A}_{\text{benign}})$, we can bound the difference between the value of $\widetilde{\pi}$ and the value of $\hat{\pi}$, and how large the different is depends on some properties of $\mathsf{Range}(\mathcal{A}_{\text{benign}})$.

Concretely, Let R and P be the reward function and transition probability function of the current agent when the other agents execute fixed policies. So $R(s,a)$ is the immediate reward of taking action a at state s, and $P(s'|s,a)$ is the probability of transitioning to state s' from s by taking action a. (Note that s is the underlying state which may not be observed by the agent.)

Definition D.7 (Dynamics Discrepancy of $\hat{\pi}$)**.** A message-ablation policy $\hat{\pi}$ is called ϵ_R,ϵ_P-discrepant if ϵ_R, ϵ_P are the smallest values such that for any $s \in \mathcal{S}$ and the corresponding benign action set $\mathcal{A}_{\text{benign}}$, we have $\forall a_1, a_2 \in \mathsf{Range}(\mathcal{A}_{\text{benign}})$,

$$|R(s,a_1) - R(s,a_2)| \leq \epsilon_R, \tag{9}$$

$$\int |P(s'|s,a_1) - P(s'|s,a_2)| \mathrm{d}s' \leq \epsilon_P. \tag{10}$$

Remarks. (1) Equation (10) is equivalent to that the total variance distance between $P(\cdot|s,a_1)$ and $P(\cdot|s,a_2)$ is less than or equal to $\epsilon_P/2$. (2) For any environment with bounded reward, ϵ_R and ϵ_P always exist.

Definition D.7 characterizes how different the local dynamics of actions in $\mathsf{Range}(\mathcal{A}_{\text{benign}})$ are, over all possible states. If $\mathsf{Range}(\mathcal{A}_{\text{benign}})$ is small and the environment is relatively smooth, then taking different actions within this range will not result in very different future rewards. The theorem below shows a reward certificate for the message-ensemble policy $\widetilde{\pi}$.

Theorem D.8 (Reward Certificate for Continuous Action Space)**.** *Let $V^{\hat{\pi}}(s)$ be the clean value (discounted cumulative reward) of $\hat{\pi}$ starting from state s under no attack; let $\tilde{V}_{\nu}^{\widetilde{\pi}}(s)$ be the value of $\widetilde{\pi}$ starting from state s under attack algorithm ν, where ν satisfies Assumption 3.1; let k be an ablation size satisfying Condition D.5. If $\hat{\pi}$ is ϵ_R,ϵ_P-discrepant, then for any state $s \in \mathcal{S}$, we have*

$$\min_{\nu} \tilde{V}_{\nu}^{\widetilde{\pi}}(s) \geq V^{\hat{\pi}}(s) - \frac{\epsilon_R + \gamma V_{\max} \epsilon_P}{1-\gamma}, \tag{11}$$

where $V_{\max} := \sup_{s,\pi} |V^{\pi}(s)|$.

Remarks. (1) The certificate holds for any attack algorithm ν with $C < \frac{N-1}{2}$. (2) If ϵ_R and ϵ_P are small, then the performance of message-ensemble policy $\widetilde{\pi}$ under attacks is similar to the performance of the message-ablation policy $\hat{\pi}$ under no attack.

It is important to note that ϵ_R and ϵ_P are intrinsic properties of $\hat{\pi}$, independent of the attacker. Therefore, one can approximately measure ϵ_R and ϵ_P during training. Similar to Condition D.2 required for a discrete action space, the gap between the attacked reward of $\widetilde{\pi}$ and the natural reward of $\hat{\pi}$ depends on how well the benign messages are reaching a consensus. (Smaller ϵ_R and ϵ_P imply that the actions in $\mathcal{A}_{\text{benign}}$ are relatively concentrated and the environment dynamics are relatively smooth.)

Moreover, one can optimize $\hat{\pi}$ during training such that ϵ_R and ϵ_P are as small as possible, to further improve the robustness guarantee of $\widetilde{\pi}$. This can be a future extension of this work.

Technical proofs of all theoretical results can be found in Appendix E.

D.3. Discussion on Ablation Size k

Our AME defense requires an extra hyperparameter: the ablation size k. In this section, we discuss the selection of k in theory and in practice.

Choosing a smaller k could make the agents more robust, but could also potentially hurt their natural performance since the agents would have access to less information shared by others. Based on the previous experimental results, it is crucial to set

k such that Condition D.5 (continuous action space) or Condition D.2 (discrete action space) is satisfied, such that agents could have performance certificates and thus remain robust under attack. Next, we analyze the relationship between k and the required conditions.

Condition D.2 and k

Different from Condition D.5 which is independent of the environment, Condition D.2 required in a discrete action space is related to the environment and the communication quality. Intuitively, Condition D.2 can be satisfied if the benign messages can reach some "consensus", i.e., there are enough purely benign k-samples voting for the same action. This can be achieved when the environment is relatively deterministic (e.g., there is a certainly optimal direction to go). Fortunately, Condition D.2 can be checked during training time, and we can train the message-ablation policy and the communication policy to increase $u_{\max}$ as much as possible.

On the other hand, Condition D.2 is also related to the selection of ablation size k. The ratio of contaminated votes is $\frac{\binom{N-1}{k}-\binom{N-1-C}{k}}{\binom{N-1}{k}}$, and it is easy to show that this rate increases as k when $k \leq N-C-1$. Therefore, when k is smaller, it is relatively easier to satisfy Condition D.2 as the adversarial messages can take over a smaller proportion of the total number of votes.

Condition D.5 and k

We first decompose Equation (7) as below.

$$\binom{N-1-C}{k} > \frac{1}{2}\binom{N-1}{k} \tag{12}$$

$$\frac{(N-1-C)!}{(N-1-C-k)!k!} > \frac{(N-1)!}{2(N-1-k)!k!} \tag{13}$$

$$1 > \frac{(N-1)\cdots(N-C)}{2(N-k-1)\cdots(N-k-C)} \tag{14}$$

Therefore, Equation (7) is equivalent to

$$(N-k-1)\cdots(N-k-C) > \frac{1}{2}(N-1)\cdots(N-C) \tag{15}$$

When $k=1$, the above inequality becomes

$$(N-2)\cdots(N-C)(N-C-1) > \frac{1}{2}(N-1)(N-2)\cdots(N-C) \tag{16}$$

$$(N-C-1) > \frac{1}{2}(N-1) \tag{17}$$

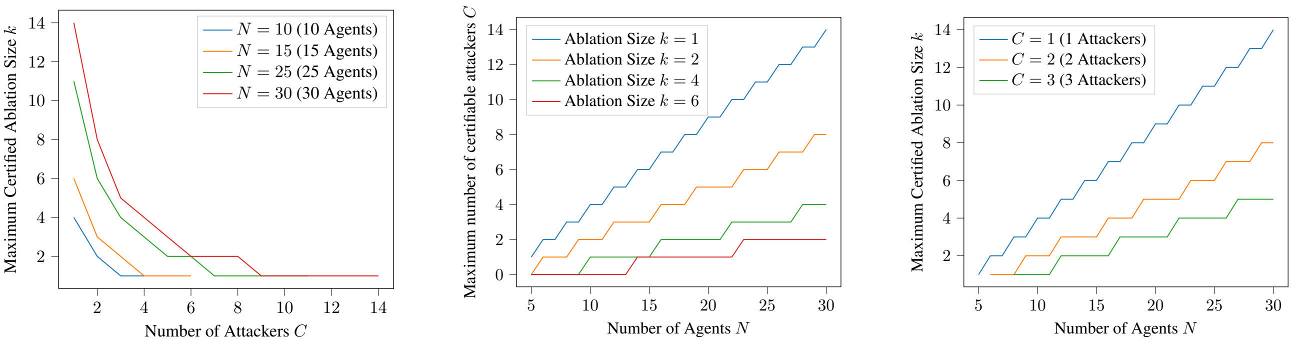
$$\frac{1}{2}(N-1) > C, \tag{18}$$

which holds under Assumption 3.1. Therefore, $\boldsymbol{k=1}$ *is always a feasible solution for Condition D.5.*

Then, when C is fixed and k goes up from 1, the LHS of Equation (15) goes down while the RHS does not change. Therefore, for a given number of agents (N) and a fixed number of adversaries (C), there exists an integer k_0 such that any $k \leq k_0$ satisfies Condition D.5. In practice, if we have an estimate of the number of adversarial messages that we would like to defend against, then we could choose the maximum k satisfying Equation (7).

On the other hand, when k is fixed, there exists a C_0 such that any $C \leq C_0$ can let Equation (15) hold. Therefore, if C is unknown, for any selection of k, Equation (15) can justify the maximum number of adversaries for the current selection.

From Equation (15), we can see the interdependence between three parameters: total number of agents N, number of adversaries C, and the ablation size k. In Figure 3 we visualize their relationship by fixing one variable at a time. From these figures, we can see an obvious trade-off between C and k — if there are more adversaries, k has to be set smaller. But when C is small, e.g. $C=1$, k can be relatively large, so the agent does not need to compromise much natural performance to achieve robustness.



(a) C vs. k **(b)** N vs. C **(c)** N vs. k

Figure 3. Relationship between the total number of agents N, number of attackers C, and ablation size k. **(a)**: Maximum certifiable ablation size k under different number of attackers. **(b)**: Maximum defensible number of attackers C with different total number of agents N. **(c)**: Maximum certifiable ablation size k with different total number of agents N.

Selecting k to Balance between Natural Performance and Robustness The above analysis shows that a smaller k makes the agent more robust, while the natural performance may be sacrificed as the message-ablation policy makes decisions based on less benign information. Such a trade-off between natural performance and robustness is common in the literature of adversarial learning (Tsipras et al., 2018; Zhang et al., 2019).

In practice, we suggest setting k to be the largest integer satisfying Equation (7). If higher robustness is needed, then k can be further decreased. If robustness is not the major concern while higher natural performance is required, one can increase k. Note that if $k = N - 1$, AME degenerates to the original vanilla policy without defense.

What If Conditions Are Not Satisfied? Even if Condition D.5 and Condition D.2 are not satisfied, the agent can still be robust under attacks as verified in our experiments (AME with $k = 2$ still achieves relatively robust performance under $C = 3$ which exceeds the theoretically largest number of certifiable adversaries). Because these conditions are needed for the certificates which consider the theoretically worst-case attacks. However, in practice, an attacker has restricted power and knowledge (e.g., it does not know the victim policy/reward, and does not know the environment dynamics as prior), and is likely to be even weaker than the learned adaptive white-box attacker we use in experiments. As a result, even if a larger k may break the conditions, it can still improve the empirical robustness of an agent in practice.

Extension: Adaptive Defense with Different k's Moreover, to allow higher flexibility, one can train multiple message-ablation policies with different selections of k's during training. Then, an adaptive strategy can be used in test time. For example, if $u_{\max}$ is too small, we can use a larger k with the corresponding trained message-ablation policy.

Extension: Gaining both Natural Performance and Robustness by Attack Detection From the analysis of the relation between N, C and k, we can see that when the number of adversaries is large, the corresponding ablation size k is supposed to be smaller. This is reasonable because a more conservative defense is needed against a stronger attacker. But if we can identify some adversarial messages and rule them out before message ablation and ensemble, we can still defend with guarantees using a relatively large k. For example, if we have identified c adversarial messages, then we only need to deal with the remaining $C - c$ adversarial messages out of $N - 1 - c$ messages. By Equation (7), a larger k can be used compared to defending C adversarial messages out of $N - 1$ messages. We also provide an adversary detection algorithm in Appendix G using a similar idea of AME.

D.4. Extension of AME: Ensemble with Partial Samples

So far we have discussed the proposed AME defense and the constructed ensemble policy that aggregates all $\binom{N-1}{k}$ number of k-samples out of $N - 1$ messages. However, if N is large, sampling all $\binom{N-1}{k}$ combinations of message subsets could be expensive. In this case, a smaller number of k-samples can be used. That is, given a sample size $0 < D \leq \binom{N-1}{k}$, we randomly select D number of k-samples from $\mathcal{H}_k(\mathbf{m})$ (without replacement), and then we aggregate the message-ablation policy's decisions on selected k-samples. In this way, we define a partial-sample version of AME, where the ensemble policy is constructed by D instead of $\binom{N-1}{k}$ samples. Let $\mathcal{H}_{k,D}(\mathbf{m})$ be a subset of $\mathcal{H}_k(\mathbf{m})$ that contains D random k-samples from $\mathcal{H}_k(\mathbf{m})$. Then the D-ensemble policy π_D is defined as

$$\widetilde{\pi}_D(\tau, \mathbf{m}) := \operatorname{argmax}_{a \in \mathcal{A}} \sum_{[\mathbf{m}]_k \in \mathcal{H}_{k,D}(\mathbf{m})} \mathbb{1}[\hat{\pi}(o, [\mathbf{m}]_k) = a], \tag{19}$$

for a discrete action space, and

$$\tilde{\pi}_D(\tau, \mathbf{m}) = \mathsf{Median}\{\hat{\pi}(\tau, [\mathbf{m}]_k)\}_{[\mathbf{m}]_k \in \mathcal{H}_{k,D}(\mathbf{m})}. \tag{20}$$

for a continuous action space.

In the partial-sample version of AME, we can still provide high-probability robustness guarantees.

For notation simplicity, let $n_1 = \binom{N-1}{k}$, $n_2 = \binom{N-C-1}{k}$. Define the majority vote as

$$u_{\max} := \max_{a \in \mathcal{A}} \sum\nolimits_{[\mathbf{m}]_k \in \mathcal{H}_{k,D}(\mathbf{m})} \mathbb{1}[\hat{\pi}(\tau, [\mathbf{m}]_k) = a]. \tag{21}$$

The following theorem shows a general guarantee for D-ensemble policy in a discrete action space.

Theorem D.9 (General Action Guarantee for Discrete Action Space)**.** *Given an arbitrary sample size $0 < D \leq \binom{N-1}{k}$, for the D-ensemble policy $\widetilde{\pi}_D$ defined in Equation* (19)*, Relation* (5) *holds deterministically if the majority vote $u_{\max} > n_1 - n_2$. Otherwise it holds with probability at least*

$$p_D = \frac{\sum_{j=0}^{u_{\max}-1} \binom{n_1 - n_2}{j} \binom{n_2}{D-j}}{\binom{n_1}{D}}. \tag{22}$$

Note that Theorem D.3 is a special case of Theorem D.9, since it assumes $u_{\max} > n_1 - n_2$.

Theorem D.10 below further shows the theoretical result for a continuous action space.

Theorem D.10 (General Action Guarantee for Continuous Action Space)**.** *Given an arbitrary sample size $0 < D \leq \binom{N-1}{k}$, for the D-ensemble policy $\widetilde{\pi}_D$ defined in Equation* (20) *with an ablation size k satisfying Condition D.5, Relation* (8) *holds with probability at least*

$$p_D = \frac{\sum_{j=\tilde{D}}^{D} \binom{n_2}{j} \binom{n_1 - n_2}{D-j}}{\binom{n_1}{D}}, \tag{23}$$

where $\tilde{D} = \lfloor \frac{D}{2} \rfloor + 1$.

The larger D is, the higher the probability p_D is, the more likely that the message-ensemble policy selects an action in $\mathsf{Range}(\mathcal{A}_{\text{benign}})$. In Theorem D.10, when $D = \binom{N-1}{k}$, the probability p_D is 1 and the result matches Theorem D.6.

Technical proofs of all theoretical results can be found in Appendix E.

E. Technical Proofs

For the simplicity of the proof, we make the following definition.

Definition E.1. (Purely Benign k-sample and contaminated k-sample) A k-sample $[\mathbf{m}]_k \in \mathcal{H}_k(\mathbf{m})$ is purely benign if every message in $[\mathbf{m}]_k$ comes from a benign agent and is unperturbed. A k-sample $[\mathbf{m}]_k \in \mathcal{H}_k(\mathbf{m})$ is contaminated if there exists some message in $[\mathbf{m}]_k$ that is perturbed.

For notation simplicity, let $n_1 := |\mathcal{H}_k(\mathbf{m})| = \binom{N-1}{k}$ be the total number of k-samples from a message set $\mathbf{m}$. Note that the total number of purely benign k-samples is $n_2 := \binom{N-C-1}{k}$, and the total number of contaminated k-samples is $n_1 - n_2 = \binom{N-1}{k} - \binom{N-C-1}{k}$.

E.1. Proofs in Discrete Action Space

Action Certificates We first prove the action certificates in the discrete action. Note that Theorem D.3 is a special case of Theorem D.9 ($u_{\max} > n_1 - n_2$ and $D = \binom{N-1}{k}$), so we first prove the general version Theorem D.9 and then Theorem D.3 holds as a result.

Proof of Theorem D.9 and Theorem D.3. The majority voted action $\tilde{a}$ is a benign action, i.e., $\tilde{a} \in \mathcal{A}_{\text{benign}}$, if the ablation policy $\hat{\pi}$ renders action $\tilde{a}$ for at least one purely benign k-sample. If $u_{\max} > n_1 - n_2$, since $n_1 - n_2$ is exactly the total number of contaminated k-samples, then it is guaranteed that there is at least one purely benign k-sample for which $\hat{\pi}$ renders $\tilde{a}$. Thus, $\tilde{a} \in \mathcal{A}_{\text{benign}}$, and Theorem D.3 holds.

On the other hand, if $u_{\max} \leq n_1 - n_2$, then in order for $\tilde{a}$ to be in $\mathcal{A}_{\text{benign}}$, among the $u_{\max}$ k-samples resulting in $\tilde{a}$ there can be at most $u_{\max} - 1$ contaminated k-samples. There are $\sum_{j=0}^{u_{\max}-1} \binom{n_1-n_2}{j}\binom{n_2}{D-j}$ such combinations in terms of the sampling of D, and the total number of combinations are $\binom{n_1}{D}$. Therefore, we get Equation (22).

□

Reward Certificate. Next, following Theorem D.3, we proceed to prove the reward certificate.

Proof of Corollary D.4. Based on the definition of benign action set $\mathcal{A}_{\text{benign}}$, $\widetilde{\pi}$ selects an action $\widetilde{a}$ if and only if there exists a purely benign k-sample $[\mathbf{m}]_k \in \mathcal{H}_k(\mathbf{m})$ such that the message-ablation policy $\hat{\pi}$ selects $\widetilde{a} = \hat{\pi}(\tau, [\mathbf{m}]_k)$. Therefore, for any trajectory generated by $\widetilde{\pi}$ under attacks, there is a trajectory of $\hat{\pi}$ with a list of k-samples $[\mathbf{m}]_k^{(1)}, [\mathbf{m}]_k^{(2)}, \cdots$ that renders the same cumulative reward under no attack.

□

E.2. Proofs in Continuous Action Space

Action Certificate. Similar to the discrete-action case, we first prove Theorem D.10, and then prove Theorem D.6 as a special case of Theorem D.10.

Proof of Theorem D.10. To understand the intuition of element-wise median operation in continuous action space, let us first start with an intuitive example: consider 5 arbitrary numbers x_1, ..., x_5, if we already know 3 of them x_1, x_2, x_3, then it is certain that $\min(x_1, x_2, x_3) \leq \mathsf{Median}(x_1, \cdots, x_5) \leq \max(x_1, x_2, x_3)$. Therefore, when purely benign k-samples form the majority (Condition D.5), the element-wise median action falls into the range of actions produced by safe messages.

To be more general, in a continuous action space, $\tilde{a} \in \mathsf{Range}(\mathcal{A}_{\text{benign}})$ is equivalent to the condition that out of the D sampled k-samples, purely benign k-samples make up the majority. There are $\sum_{j=\tilde{D}}^{D} \binom{n_2}{j}\binom{n1-n2}{D-j}$ such combinations in terms of the sampling of D, where $\tilde{D} = \lfloor \frac{D}{2} \rfloor + 1$. Once again the total number of combinations is $\binom{n_1}{D}$. Therefore, we get Equation (23). □

Proof of Theorem D.6. The proof of Theorem D.6 follows as a special case of Theorem D.10 when $D = \binom{N-1}{k} = n_1$. In this case, the only non-zero term left in the numerator of p_D is $\binom{n_2}{j}\binom{n_1-n_2}{n_1-j} = \binom{n_2}{n_2}\binom{n_1-n_2}{n_1-n_2} = 1$ (we need $n_2 \geq j$ and $n_1 - n_2 \geq n_1 - j$ to keep the numerator from vanishing, which implies $j = n_2$, which is no lower than $\tilde{D}$ since $n_2 > n_1/2$ due to Condition D.5). Hence we have $p_D = 1$. □

Reward Certificate. Next, we derive the reward guarantee for the continuous-action case.

Proof of Theorem D.8. We let $\mathbb{P}(a|s;\pi)$ be the probability of the message-ablation policy $\hat{\pi}$ taking action a at state s, where π can be either the message-ablation policy $\hat{\pi}$ or the message-ensemble policy $\widetilde{\pi}$. Note that this is a conditional probability function, and the policy does not necessarily observe s.

Without loss of generality, let ν^* be the optimal attacking algorithm that minimizes $\tilde{V}_{\nu}^{\widetilde{\pi}}$. Let $\mathcal{A}_s$ denote the range of benign action at state s induced by the current message-ablation policy $\hat{\pi}$. Then we have

$$\begin{aligned}
&\sup_{s\in\mathcal{S}}\left|V^{\hat{\pi}}(s)-\tilde{V}^{\tilde{\pi}}_{\nu^*}(s)\right| \\
&=\sup_{s\in\mathcal{S}}\left|\mathbb{E}_{a\sim\mathbb{P}(a|s;\hat{\pi})}\left[R(s,a)+\gamma\int P(s'|s,a)V^{\hat{\pi}}(s')\mathrm{d}s'\right]-\mathbb{E}_{a\sim\mathbb{P}(a|s;\tilde{\pi})}\left[R(s,a)+\gamma\int P(s'|s,a)\tilde{V}^{\tilde{\pi}}_{\nu^*}(s')\mathrm{d}s'\right]\right| \\
&\le\sup_{s\in\mathcal{S}}\sup_{a_1,a_2\in\mathcal{A}_s}\left|R(s,a_1)+\gamma\int P(s'|s,a_1)V^{\hat{\pi}}(s')\mathrm{d}s'-R(s,a_2)-\gamma\int P(s'|s,a_2)\tilde{V}^{\tilde{\pi}}_{\nu^*}(s)(s')\mathrm{d}s'\right| \\
&\le\sup_{s\in\mathcal{S}}\sup_{a_1,a_2\in\mathcal{A}_s}|R(s,a_1)-R(s,a_2)|+\sup_{a_1,a_2\in\mathcal{A}_s}\left|\gamma\int P(s'|s,a_1)V^{\hat{\pi}}(s')\mathrm{d}s'-\gamma\int P(s'|s,a_2)\tilde{V}^{\tilde{\pi}}_{\nu^*}(s')\mathrm{d}s'\right| \\
&\le\epsilon_R+\gamma\sup_{s\in\mathcal{S}}\sup_{a_1,a_2\in\mathcal{A}_s}\left|\int P(s'|s,a_1)V^{\hat{\pi}}(s')\mathrm{d}s'-\int P(s'|s,a_2)\tilde{V}^{\tilde{\pi}}_{\nu^*}(s')\mathrm{d}s'\right| \qquad (24)\\
&\le\epsilon_R+\gamma\sup_{s\in\mathcal{S}}\sup_{a_1,a_2\in\mathcal{A}_s}\left|\int P(s'|s,a_1)V^{\hat{\pi}}(s')\mathrm{d}s'-\int P(s'|s,a_1)\tilde{V}^{\tilde{\pi}}_{\nu^*}(s')\mathrm{d}s'\right| \\
&\qquad+\gamma\sup_{s\in\mathcal{S}}\sup_{a_1,a_2\in\mathcal{A}_s}\left|\int P(s'|s,a_1)\tilde{V}^{\tilde{\pi}}_{\nu^*}(s')\mathrm{d}s'-\int P(s'|s,a_2)\tilde{V}^{\tilde{\pi}}_{\nu^*}(s')\mathrm{d}s'\right| \\
&\le\epsilon_R+\gamma\sup_{s\in\mathcal{S}}\left|V^{\hat{\pi}}(s)-\tilde{V}^{\tilde{\pi}}_{\nu^*}(s)\right|+\gamma\left|\int P(s'|s,a_1)\tilde{V}^{\tilde{\pi}}_{\nu^*}(s')\mathrm{d}s'-\int P(s'|s,a_2)\tilde{V}^{\tilde{\pi}}_{\nu^*}(s')\mathrm{d}s'\right| \\
&\le\epsilon_R+\gamma\sup_{s\in\mathcal{S}}\left|V^{\hat{\pi}}(s)-\tilde{V}^{\tilde{\pi}}_{\nu^*}(s)\right|+\gamma V_{\max}\epsilon_P.
\end{aligned}$$

By solving for the recurrence relation over $\sup_{s\in\mathcal{S}}\left|V^{\hat{\pi}}(s)-\tilde{V}^{\tilde{\pi}}_{\nu^*}(s)\right|$, we obtain

$$\sup_{s\in\mathcal{S}}\left|V^{\hat{\pi}}(s)-\tilde{V}^{\tilde{\pi}}_{\nu^*}(s)\right|\le\frac{\epsilon_R+\gamma V_{\max}\epsilon_P}{1-\gamma}. \tag{25}$$

which leads to the desired relation in Theorem D.8.

□

F. Full Experiment Results and Details

Section F.1, F.2 and F.3 provide experimental details and full empirical results in FoodCollector, InventoryManager and MARL-MNIST, respectively. Section F.4 shows hyperparamter tests of AME.

F.1. Details and Full Results in FoodCollector Environment

F.1.1. Environment Description

The FoodCollector environment is a 2D particle world shown by Figure 4(left). There are $N = 9$ agents with different colors, and N foods with colors corresponding to the N agents. Agents are rewarded when eating foods with the same color. A big round obstacle is located in the center of the map, which the agent cannot go through. There are some poisons (shown as black dots) in the environment, and the agents get penalized whenever they touch the poison. Each agent has 6 sensors that detect the objects around it, including the poisons and the colored foods. The game is episodic, with horizon set to be 200. In the beginning of each episode, the agents, foods and poisons are randomly generated in the world.

State Observation Each of the 6 sensors can detect the following values when the corresponding element is within the sensor's range (the corresponding dimensions are 0's if nothing is detected):
(1) (if detects a food) the distance to a food (real-valued);
(2) (if detects a food) the color of the food (one-hot);
(3) (if detects an obstacle) the distance to the obstacle (real-valued);
(4) (if detects the boundary) the distance to the boundary (real-valued);
(5) (if detects another agent) the distance to another agent (real-valued).

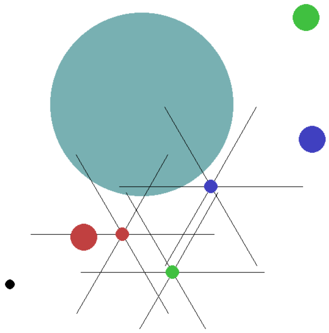


Figure 4. The FoodCollector Environment. For figure readability, we only show 3 agents colored as red/green/blue. In our experiments, there are 9 agents.

The observation of an agent includes an agent-identifier (one-hot encoding), its own location (2D coordinates), its own velocity, two flags of colliding with its food and colliding with poison, and the above sensory inputs. Therefore, the observation space is a $(7N + 30)$-dimensional vector space.

Agent Action The action space can be either *discrete* or *continuous*. For the discrete version, there are 9 actions including 8 moving directions (north, northwest, west, southwest, south, southeast, east, northeast) and 1 no-move action. For the continuous version, the action is an acceleration decision, denoted by a 2-dimensional real-valued vector, with each coordinate taking values in $[-0.01, 0.01]$.

Reward Function At every step, each agent will receives a negative reward -0.5 if it has not eaten all its food. In addition, it receives extra -1 reward if it collides with a poison. Therefore, every agent is expected to explore the environment and eat all food as fast as possible. The team reward is calculated by the average of all agents' local rewards. Note that the agents' actions do not affect each other, because they have different target foods to collect. *Agents collaborate only via communication introduced below.*

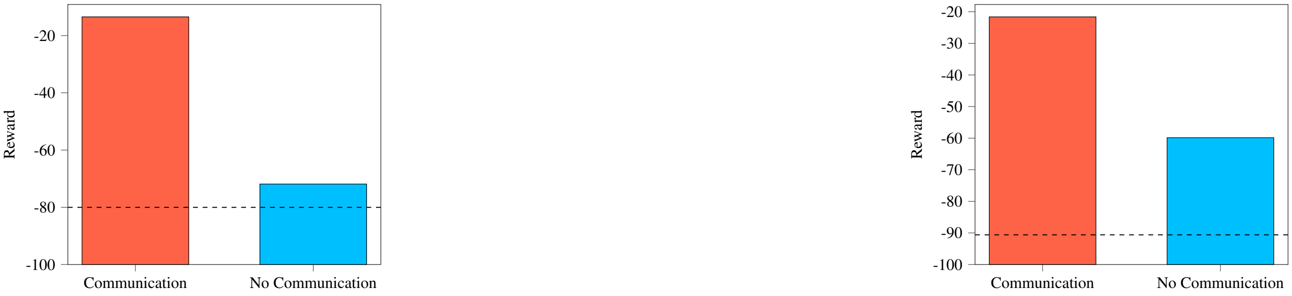
Communication Protocol Due to the limited sensory range, every agent can only see the objects around it and thus only partially observes the world. Therefore, communication among agents can help them find their foods much faster. Since our focus is to defend against adversarially perturbed communications, we first define a valid and beneficial communication protocol, where an agent sends a message to a receiver once it observes a food with the receiver's color. For example, if a red agent encounters a blue food, it can then send a message to the blue agent so that the blue agent knows where to find its food. To remember the up-to-date communication, every agent maintains a list of most recent $N - 1$ messages sent from other $N - 1$ agents. A message contains the sender's current location and the relative distance to the food (recorded by the 6 sensors), which are bounded between -1 and 1. Therefore, a message is a 8-dimensional vector, and each agent's communication list has $8(N - 1)$ dimensions in total.

Communication Gain During training with communication, we concatenate the observation and the communication list of the agent to an MLP-based policy, compared to the non-communicative case where the policy only takes in local observations. More implementation details are in Appendix F.1.2. As verified in Figure 5, communication does help the agent to obtain a much higher reward in both discrete action and continuous action cases, which suggests that the agents tend to rely heavily on the communication messages for finding their food.

F.1.2. Implementation Details

Implementation of Trainer In our experiments, we use the Proximal Policy Optimization (PPO) (Schulman et al., 2017) algorithm to train all agents (with parameter sharing among agents) as well as the attackers. Specifically, we adapt from the elegant OpenAI Spinning Up (Achiam, 2018) Implementation for PPO training algorithm. On top of the Spinning Up PPO implementation, we also keep track of the running average and standard deviation of the observation and normalize the observation. All experiments are conducted on NVIDIA GeForce RTX 2080 Ti GPUs.

For the policy network, we use a multi-layer perceptron (MLP) with two hidden layers of size 64. For a discrete action space, a categorical output distribution is used. For a continuous action space, since the valid action is bounded within a



(a) Discrete action space

(b) Continuous action space

Figure 5. **FoodCollector**: Reward of agents trained by PPO with communication v.s. without communication. Black dashed line stands for the mean performance of the agent when selecting actions uniformly randomly.

small range [-0.01,0.01], we parameterize the policy as a Beta distribution, which has been proposed in previous works to better solve reinforcement learning problems with bounded actions (Chou et al., 2017). In particular, we parameterize the Beta distribution by α_θ and β_θ, such that $\alpha = \log(1 + e^{\alpha_\theta(s)}) + 1$ and $\beta = \log(1 + e^{\beta_\theta(s)}) + 1$ (1 is added to make sure that $\alpha, \beta \geq 1$). Then, $\pi(a|s) = f(\frac{a-h}{2h}; \alpha, \beta)$, where $h = 0.01$, and $f(x; \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$ is the density function of the Beta Distribution. For the value network, we also use an MLP with two hidden layers of size 64.

In terms of other hyperparameters used in the experiments, we use a learning rate of 0.0003 for the policy network, and a learning rate of 0.001 is used for the value network. We use the Adam optimizer with $\beta_1 = 0.99$ and $\beta_2 = 0.999$. For every training epoch, the PPO agent interacts with the environment for 4000 steps, and it is trained for 500 epochs in our experiments.

Implementation of Attackers An attacker maps its own observation to the malicious communication messages that it will send to the victim agent. Thus, the action space of the attacker is the communication space of a benign agent, which is bounded between -1 and 1.

- *Non-adaptive Attacker* We implement a fast and naive attacking method for the adversary. At every dimension, the naive attacker randomly picks 1 or -1 as its action, and then sends the perturbed message which consists entirely of 1 or -1 to the victim agent.

- *Adaptive RL Attacker* We use the PPO algorithm to train the attacker, where we set the reward of the attacker to be the negative reward of the victim. The attacker uses a Gaussian policy, where the action is clipped to be in the valid communication range. The network architecture and all other hyperparameter settings follow the exact same from the clean agent training.

Implementation of Baselines

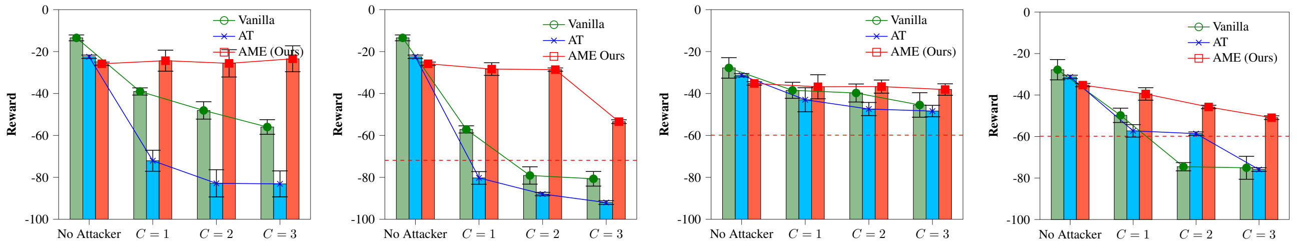
- *Vanilla Learning* For Vanilla method, we train a shared policy network to map observations and communication to actions.

- *Adversarial Training (AT)* For adversarial training, we alternate between the training of attacker and the training of the victim agent. Both the victim and attackers are trained by PPO. For every 200 training epochs, we switch the training, where we either fix the trained victim and train the attacker for the victim or fix the trained attacker and train the victim under attack. We continue this process for 10 iterations.

Note that the messages are symmetric (of the same format), we shuffle the messages before feeding them into the policy network for both Vanilla and AT, to reduce the bias caused by agent order. We find that shuffling the messages helps the agent converge much faster (50% fewer total steps). Note that AME randomly selects k-samples and thus messages are also shuffled.

F.1.3. Additional Results

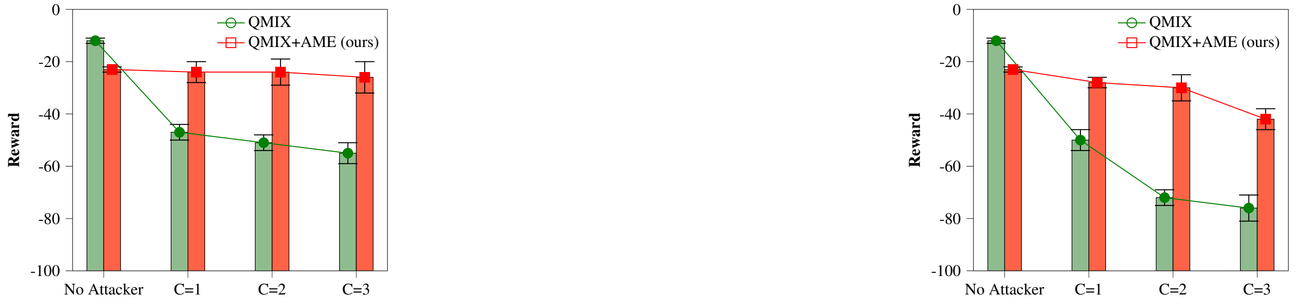
Robust Performance We show the performance of AME and baselines in discrete-action and continuous-action FoodCollectors in Figure 6.



(a) Disc. & Non-adaptive **(b)** Disc. & Adaptive **(c)** Cont. & Non-adaptive **(d)** Cont. & Adaptive

Figure 6. Rewards of our AME and baselines in discrete-action and continuous-action FoodCollector, under no attacker and varying numbers of adversaries for adaptive and various non-adaptive attacks. The dashed red lines stand for the average performance of a non-communicative agent. Results are averaged over 5 random seeds.

Additional Results with QMIX AME is a generic defense approach that can be used for any RL/MARL learning algorithm. In Figure 7, we show the results of AME combined with an MARL algorithm QMIX (Rashid et al., 2018) in the discrete-action FoodCollector environment (QMIX does not work for continuous actions so it is not applicable in continuous-action FoodCollector and InventoryManager). Compared to the vanilla QMIX algorithm, QMIX+AME achieves much higher robustness and stable performance under various numbers of adversaries.



(a) Discrete & Non-adaptive **(b)** Discrete & Adaptive

Figure 7. Reward comparison between original QMIX and QMIX combined with our AME in FoodCollector with discrete action space, under no attacker or non-adaptive/adaptive attacks under varying numbers of adversaries. For AME, the ablation size k is set as 2.

F.2. Details and Full Results in InventoryManager Environment

F.2.1. Environment Description

The *InventoryManager* environment is an inventory management setup, where $N = 10$ cooperative heterogeneous distributors carry inventory for $M = 3$ products. A population of $B = 300$ buyers request a product from a randomly selected distributor agent according to a demand distribution $\mathbf{p} = [p_1, \dots, p_M]$. We denote the demand realization for distributor i with $\mathbf{d_i} = [d_{i,1}, \dots, d_{i,M}]$. Distributor agents manage their inventory by restocking products through interacting with the buyers. The game is episodic with horizon set to 50. At the beginning of each episode, a realization of the demand distribution $\mathbf{p}$ is randomly generated and the distributors' inventory for each product is randomly initialized from $[0, \frac{B}{N}]$, where $\frac{B}{N}$ is the expected number of buyers per distributor. The distributor agents are penalized for mismatch between their inventory and the demand for a product, and they aim to restock enough units of a product at each step to prevent insufficient inventory without accruing a surplus at the end of each step.

State Observation A distributor agent's observation includes its inventory for each product, and the products that were requested by buyers during the previous step. The observation space is a $2M$-dimensional vector.

Agent Action Distributors manage their inventory by restocking new units of each product or discarding part of the leftover inventory at the beginning of each step. Hence, agents take both positive and negative actions denoted by an

M-dimensional vector, and the action space can be either discrete or continuous. In our experiments, we use continuous actions assuming that products are divisible and distributors can restock and hold fractions of a product unit.

Reward Function During each step, agent i's reward is defined as $r_i = -|| \max(\mathbf{I}_i + \mathbf{a}_i, 0) - \mathbf{d}_i||_2$, where $\mathbf{I}_i$ denotes the agents initial inventory vector, and $\mathbf{a}_i$ denotes the inventory restock vector from action policy π_i. Note that the agents' actions do not affect each other. *Agents collaborate only via communication introduced below.*

Communication Protocol Distributors learn the demand distribution and optimize their inventory by interacting with their own customers (i.e., portion of buyers that request a product from that distributor). Distributors would benefit from sharing their observed demands with each other, so that they could estimate the demand distribution more accurately for managing inventory. At the end of each step, a distributor communicates an M-dimensional vector reporting its observed demands to all other agents. In the case of adversarial communications, this message may differ from the agents' truly observed demands.

Communication Gain During training with communication, messages received from all other agents are concatenated to the agent's observation, which is used to train an MLP-based policy, compared to the non-communicative case where the policy is trained using only local observations. As observed from Figure 8, communication helps agents obtain higher rewards since they are able to manage their inventory based on the overall demands observed across the population of buyers rather than their local observation. Results are reported by averaging rewards corresponding to 5 experiments run with different training seeds.

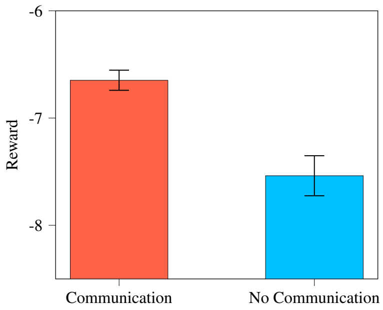


Figure 8. **InventoryManager**: Reward of agents trained with communication v.s. without communication.

F.2.2. Implementation Details

Implementation of Trainer As in the FoodCollector experiments, we use the PPO algorithm to train all agent action policy as well as adversarial agent communication policies. We use the same MLP-based policy and value networks as the FoodCollector but parameterize the policy as a Gaussian distribution. The PPO agent interacts with the environment for 50 steps, and it is trained for 10000 episodes. The learning rate is set to be 0.0003 for the policy network, and 0.001 for the value network.

Implementation of Attackers The attacker uses its observations to communicate malicious messages to victim distributors, and its action space is the communication space of a benign agent. We consider the following non-adaptive and adaptive attackers:

- *Non-adaptive Attacker*: The attacker's goal is to harm the victim agent by misreporting its observed demands so that the victim distributor under-estimates or over-estimates the restocking of products. In our experiments, we evaluate the effectiveness of defense strategies against the following attack strategies:
  - Perm-Attack: The communication message is a random permutation of the true demand vector observed by the attacker.
  - Swap-Attack: In order to construct a communication vector as different as possible from its observed demand, the attacker reports the most requested products as the least request ones and vice versa. Therefore, the highest demand among the products is interchanged with the lowest demand, the second highest demand is interchanged with the second lowest demand and so forth.
  - Flip-Attack: Adversary i modifies its observed demand $\mathbf{d}_i$ by mirroring it with respect to $\eta = \frac{1}{M}\sum_{j=1}^{M} d_{ij}$,

such that products demanded less than η are reported as being requested more, and conversely, highly demanded products are reported as less popular.

- *Adaptive Attacker*: The attacker communication policy is trained using the PPO algorithm, and its reward is set as the negative reward of the victim agent. The attacker uses a Gaussian policy with a softmax activation in the output layer to learn a adversarial probability distribution across products, which is then scaled by the total observed demands $\sum_{j=1}^{M} d_{ij}$ to construct the communication message.

Implementation of Baselines

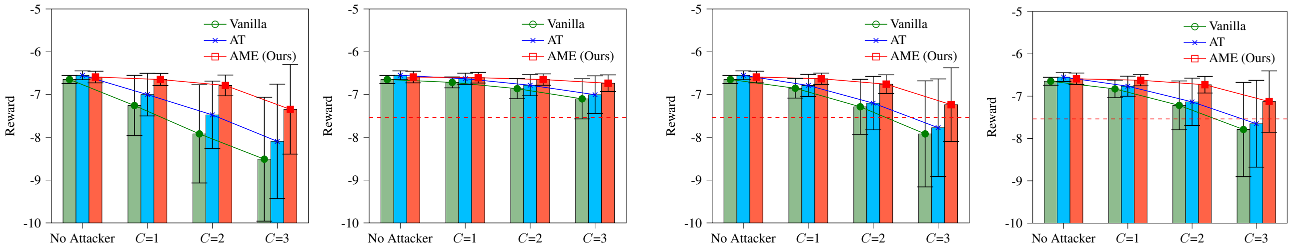
- *Vanilla Learning*: In the vanilla training method with no defense mechanism against adversarial communication, we train a shared policy network using agents' local observations and their received communication messages.

- *Adversarial Training (AT)*: We alternate between training the agent action policy and the victim communication policy, both using PPO. For the first iteration we train the policies for 10000 episodes, and then use 1000 episodes for 5 additional training alteration iterations for both the action policy and the adversarial communication policy. For more efficient adversarial training, we first shuffle the received communication messages before feeding them into the policy network. Consequently, the trained policy treats communications received from different agents in a similar manner, and we are able to only train the policy with a fixed set of adversarial agents rather than training the network on all possible combinations of adversarial agents.

For Vanilla and AT, we do not shuffle the communication messages being input to the policy, as we did not observe improved convergence, as in the FoodCollector environment.

F.2.3. Additional Results

Robust Performance The performance of AME and baselines under various attacks is shown in Figure 9.



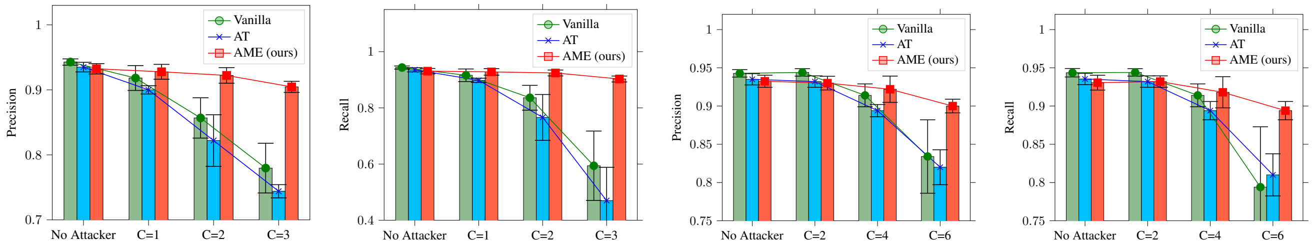
(a) Adaptive Attack **(b)** Perm-Attack **(c)** Swap-Attack **(d)** Flip-Attack

Figure 9. Rewards of our AME and baselines in InventoryManager, under no attacker and varying numbers of adversaries for adaptive and various non-adaptive attacks. The dashed red lines stand for the average performance of a non-communicative agent. Results are averaged over 5 random seeds.

F.3. Details and Full Results in MARL-MNIST Environment

F.3.1. Environment Description

We use the environment setup proposed by Mousavi et al. (2019), where agents collaboratively classify an unknown image by their observations and inter-agent communication. More specifically, we use $N = 9$ agents in the MNIST dataset of handwritten digits (LeCun et al., 1998). The dataset consists of 60,000 training images and 10,000 testing images, where each image has 28×28 pixels. There are $h = 5$ steps in an episode. In the beginning, all agents start from a pre-determined spatial configuration. At every step, each agent observes a local 5×5 patch, performs some local data processing, and shares the result with neighboring agents (we use a fully connected communication graph). With given observation and communication, each agent outputs an action in $\{Up, Down, Left, Right\}$. By each movement, the agent is translated in the desired direction by 5 pixels. In the end of an episode, agents make predictions, and all of them are rewarded by $-\text{prediction\_loss}$.



(a) Adaptive Attack **(b)** Adaptive Attack **(c)** Non-adaptive Attack **(d)** Non-adaptive Attack

Figure 10. **(MARL-MNIST)**: Precision and recall of MARL classification on MNIST without or with AME, under learned adaptive attacks and non-adaptive random attacks. All results are averaged over 5 random seeds.

F.3.2. Implementation Details

We use the same network architecture and hyperparameter setting as Mousavi et al. (2019), which are implemented in (Berrien, 2019).

Network Architecture Concretely, at every step, the input of every agent contains 3 components: (1) an encoded observation with two convolutional layers followed by vectorization and a fully connected layer; (2) the average of all communication messages from other agents; (3) a position encoding computed by feeding the current position into a single linear layer. These 3 components are concatenated and passed to two independent LSTM modules, one is for an acting policy, another is for a message generator. In the end of an episode, every agent uses its final cell state to generate a prediction using a 2-layer MLP. Then we take the average of the output logits of all agents, and use a softmax function to obtain the final probabilistic label prediction. The reward is the opposite number of the L2 difference between the prediction and the one-hot encoding of true image label.

Hyperparameters In our experiments, we follow the default hyperparameter setting in (Berrien, 2019). We use $N = 9$ agents. The size of LSTM belief state is 128. The hidden layers have size 160. The message size is set to be 32. The state encoding has size 8. We use an Adam optimizer with learning rate 1e-3. We train the agents in the MNIST dataset for 40 epochs.

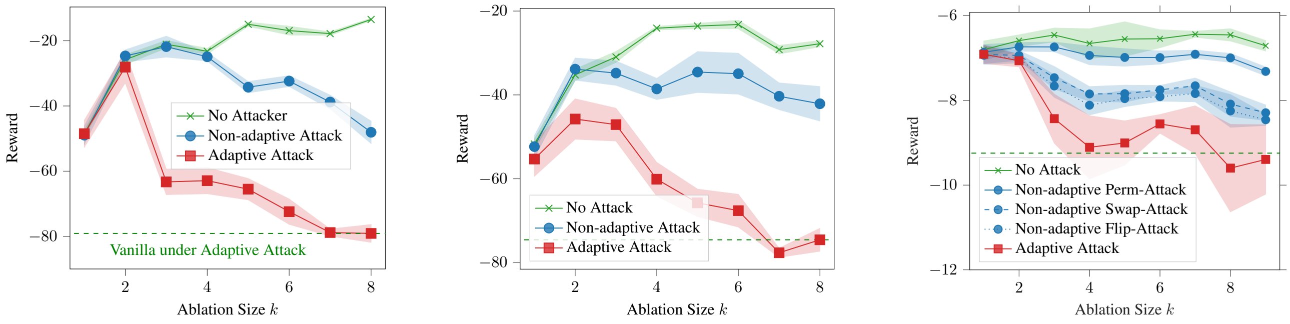
Attackers Since the communication messages are learned by neural networks, we perturb the C messages received by each agent. To make sure that the messages are not obviously detectable, we clip every dimension of the perturbed message into the range of $[-3, 3]$. The non-adaptive attacker randomly generates a new message. The adaptive attacker learns a new message generator based on its own belief state, which is trained with learning rate 1e-3 for 50 epochs.

F.3.3. Additional Results

Figure 10 demonstrates the robust performance of the MARL algorithm proposed by Mousavi et al. (Mousavi et al., 2019) with our AME defense or baseline defenses (Vanilla and AT). We set $N = 9$ and $k = 2$ for all experiments. Under learned adaptive attackers, the original MARL classifier (Vanilla) (Mousavi et al., 2019) without AME suffers from significant performance drop in terms of both precision and recall. Defending with adversarial training (AT) does not achieve good robustness, either. But AME considerably improves the robustness of agents across different numbers of attackers.
Under random attacks, we find that the original MARL classifier (Mousavi et al., 2019) is moderately robust when a few communication signals are randomly perturbed. However, when noise exists in many communication channels (e.g. $C = 6$), the performance decreases a lot. In contrast, our AME still achieves high performance when $C = 6$ communication messages are corrupted, even if the guarantee of ablation size k only holds for $C \leq 2$. Therefore, we again emphasize the the theoretical guarantee considers the worst-case strong attack, while under a relatively weak attack, we can achieve better robustness beyond what the theory suggests.

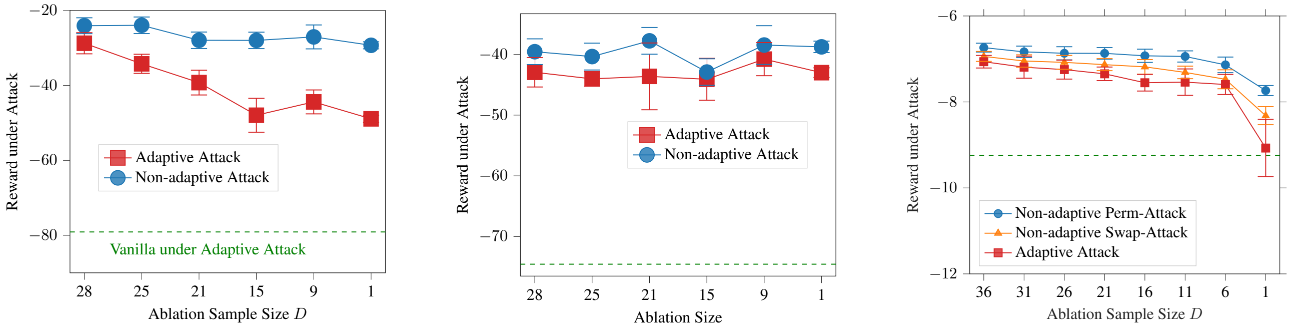
F.4. Hyperparameter Tests

Hyperparameter Tests for Ablation Size k To see the empirical influence of k on AME, we evaluate AME's natural reward and reward under $C = 2$ attacks with different ablation sizes k. The results in multiple environments are shown in Figure 11. We observe that a larger k leads to higher natural performance since each agent could gather more information from others. However, raising k also increases the risk of making decisions based on communication messages sent from



(a) Discrete-action FoodCollector **(b)** Continuous-action FoodCollector **(c)** InventoryManager

Figure 11. Natural and robust performance of AME with various values of ablation size k under $C = 2$. Dashed green lines refer to the performance of Vanilla agent under $C = 2$ attacks.



(a) Discrete-action FoodCollector **(b)** Continuous-action FoodCollector **(c)** InventoryManager

Figure 12. Natural and robust performance of AME with various values of sample size D under $C = 2$. Dashed green lines refer to the performance of Vanilla agent under $C = 2$ attacks.

an adversary. Therefore, *increasing ablation size k trades off robustness for natural performance*, matching the analysis in Appendix D.3. Moreover, as the largest solution to Equation (7) in Condition D.5, ablation size $k = 2$ achieves a good balance between performance and robustness. Even when $k > 2$ which breaks Condition D.5, AME is still more robust than baselines.

Hyper-parameter Test for Sample Size D We further evaluate the partial-sample variant of AME introduced in Appendix D.4 using $k = 2$ under $C = 2$ adversaries, with D varying from $\binom{N-1}{k}$ (the largest sample size) to 1 (the smallest sample size). Figure 12 demonstrates the performance of different D's in multiple environments. As D goes down, AME obtains lower reward under attackers, but it is still significantly more robust than baseline methods. For example, AME in FoodCollector obtains much higher reward than Vanilla under attacks even when $D = 1$ (using only 1 random k-sample in the message-ensemble policy), which is both computationally efficient and robust.

G. Discussion: Detecting Malicious Messages with Ablation

As discussed in Appendix D.3, to defend against a larger C, one has to choose a relatively small k for certifiable performance. However, Figure **??** suggests that a small k also sacrifices the natural performance of the agent to obtain higher robustness. This is known as the trade-off between robustness and accuracy (Tsipras et al., 2018; Zhang et al., 2019). Can we achieve better robustness while not sacrificing much natural performance, or obtain higher natural reward while not losing robustness?

We point out that with our proposed AME defense, it is possible to choose a larger k than what is required by Condition D.5 without sacrificing robust performance by identifying the malicious messages beforehand. The idea is to detect the adversarial messages and to rule out them before message ablation and ensemble, during the test time.

We hypothesize that given a well-trained victim policy, malicious messages tend to mislead the victim agent to take an action that is ”far away” from a ”good” action that the victim is supposed to take. To verify this hypothesis, we first train a message-ablation policy $\hat{\pi}_i$ with $k = 1$ for agent i. Then for every communication message that another agent j sends, we compute the action a_j that the victim policy $\hat{\pi}_i$ chooses based on all message subsets containing message $m_{j\to i}$, i.e. $a_j = \hat{\pi}(\tau_i, m_{j\to i})$ (note that $k = 1$ so $\hat{\pi}$ only takes in one message at a time). We then define the *action bias* as $\beta_j = \|a_j - \mathsf{Median}\{a_k\}_{k=1}^N\|_1$. Based on our hypothesis, an agent which has been hacked by attackers should induce a significantly larger action bias since they are trying to mislead the victim to take a completely different action. Here, we execute the policy of two hacked agents together with six other good agents for twenty episodes and calculate the average action bias for each agent. As shown in in Figure 13, the agents hacked by attackers indeed induce a larger action bias compared to other benign agents, which suggests the effectiveness of identifying the malicious messages by action bias.

After filtering out c messages, one could compute the required k by Equation (7) based on $C - c$ malicious messages and $N - 1 - c$ total messages, which can be larger than the largest k induced by C malicious messages out of $N - 1$ total messages. For example, if $N = 30$, $C = 3$, then the largest k satisfying Equation (7) is 5, but when 1 adversarial message is filtered out, the largest k that can defend against the remaining 2 adversarial messages is 8. Although the adversary identification is not theoretically guaranteed to be accurate, Figure 13 demonstrates the effectiveness of the adversarial message detection, which, combined with AME with larger k's, has the potential to achieve high natural performance and strong robustness in practice.

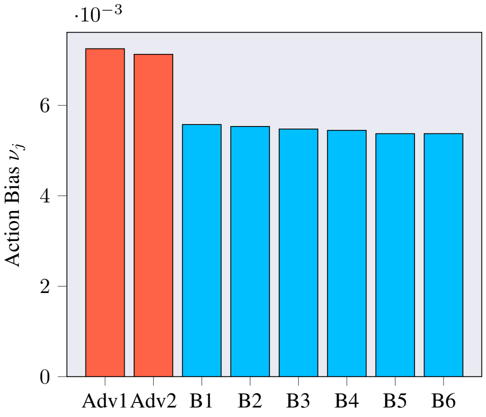


Figure 13. Attacker identification based on action bias ν_j. Adv1 and Adv2 stands for two agents hacked by the attacker. B1 up to B6 stands for six other benign agents.