RGMComm: Return Gap Minimization via Discrete Communications in Multi-Agent Reinforcement Learning

Jingdi Chen[1], **Tian Lan**[1], **Carlee Joe-Wong**[2]
[1]The George Washington University
[2]Carnegie Mellon University
jingdic@gwu.edu, tlan@gwu.edu, cjoewong@andrew.cmu.edu

Abstract

Communication is crucial for solving cooperative Multi-Agent Reinforcement Learning tasks in partially observable Markov Decision Processes. Existing works often rely on black-box methods to encode local information/features into messages shared with other agents, leading to the generation of continuous messages with high communication overhead and poor interpretability. Prior attempts at discrete communication methods generate one-hot vectors trained as part of agents' actions and use the Gumbel softmax operation for calculating message gradients, which are all heuristic designs that do not provide any quantitative guarantees on the expected return. This paper establishes an upper bound on the return gap between an ideal policy with full observability and an optimal partially observable policy with discrete communication. This result enables us to recast multi-agent communication into a novel online clustering problem over the local observations at each agent, with messages as cluster labels and the upper bound on the return gap as clustering loss. To minimize the return gap, we propose the Return-Gap-Minimization Communication (RGMComm) algorithm, which is a surprisingly simple design of discrete message generation functions and is integrated with reinforcement learning through the utilization of a novel Regularized Information Maximization loss function, which incorporates cosine-distance as the clustering metric. Evaluations show that RGMComm significantly outperforms state-of-the-art multi-agent communication baselines and can achieve nearly optimal returns with few-bit messages that are naturally interpretable.

Introduction

In multi-agent tasks, communication is necessary to successfully complete tasks when agents have partial observability of the environment. Multi-agent reinforcement learning (MARL) has recently seen success in scenarios that require communication (Cao et al. 2013). Existing approaches to multi-agent communications have considered *continuous communication* (Foerster et al. 2016; Sukhbaatar, Szlam, and Fergus 2016; Lowe et al. 2017; Jiang and Lu 2018; Wang et al. 2019; Rangwala and Williams 2020) and *discrete communication* (Mordatch and Abbeel 2017; Freed et al. 2020; Lazaridou and Baroni 2020; Li et al. 2022; Tucker et al. 2022).

However, *continuous communication messages* refer to numerical vectors originating from a continuous space. These vectors are generated by encoding local information/features using Deep Neural Networks (DNNs) or attention networks. This is largely a black-box approach with high communication overhead and offers little explainability of the messages (Chen et al. 2023). Efforts on generating *discrete communication messages* enable agents to broadcast one-hot vectors apt for solving particular tasks. However, discretizing messages through the imposition of one-hot vectors, as outlined in (Lowe et al. 2017; Sukhbaatar, Szlam, and Fergus 2016; Freed et al. 2020; Lazaridou and Baroni 2020; Li et al. 2022), inherently precludes agents from learning some desirable properties of the messages. This is because the one-hot vectors do not establish relationships between messages, since each one-hot vector remains orthogonal to and equally distant from all other one-hot vectors. Some recent works try to address the limitations of one-hot communication by endowing agents with learnable messages (Tucker et al. 2021) or discretized bottleneck layer outputs of autoencoders and using them as communication vectors (Tucker et al. 2022), but none of these methods provides average return guarantees for a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) with communication, and the communication module in (Tucker et al. 2022) still requires a large vocabulary size.

In this paper, we propose a discrete MARL communication framework, Return-Gap-Minimization Communication (RGMComm), that is proven to achieve a closed-form guarantee on optimal expected average return for Dec-POMDPs. It analyzes what information in an agent's local observations is relevant for other agents' decision-making (for minimizing the resulting return gap) and supports sharing messages between MARL agents using any finite-size discrete alphabet. More precisely, consider a Dec-POMDP problem with n agents. Each agent j computes a message m_j sent to each other agent i based on its local observation o_j in each step t. Let $\boldsymbol{m_{-i}} = \{m_j, \forall j \neq i\}$ be the collection of all messages received by agent i. For any discrete communication strategy, we quantify the gap between the optimal expected average return of an ideal policy with full observability, i.e., $\pi^* = [\pi_i^*(a_i|o_1, \ldots, o_n), \forall i]$ and the optimal expected average return of a communication-enabled, partially-observable policy, i.e., $\pi = [\pi_i(a_i|o_i, \boldsymbol{m_{-i}}), \forall i]$. This return gap is

proven to be bounded by $O(\sqrt{\epsilon}nQ_{\max})$, with respect to the number of agents n, the highest action-value $Q_{\max}$ and the average cosine-distance ϵ between joint action-value vectors corresponding to the same messages.

To the best of our knowledge, this is the first theoretical result quantifying the value of communication in POMDP through closed-form return bound. Our key insight is that the problem of generating messages m_j from local observations o_j can be viewed as an online clustering problem with o_j as inputs and m_j as labels, with respect to a new loss function to minimize the return bound. Since the policy of agent i with communication is conditioned on local observation o_i and messages $\boldsymbol{m_{-i}}$ rather than the full observation $o_1, \ldots, o_n$, its impact on optimal expected average return can be characterized using a Policy Change Lemma and relying on the average distance for corresponding joint action-values in each cluster. Intuitively, if the action values in each cluster are likely maximized at the same actions, then it would result in only small return gaps. We formalize this argument and derive an upper bound on the return gap in this paper.

The result naturally motivates a new family of MARL by training an online clustering network for message label generation. Since joint action values are available in most state-of-the-art MARL algorithms adopting actor-critic framework (Chen, Wang, and Lan 2021; Mei et al. 2023; Chen, Lan, and Choi 2023), e.g., MADDPG(Lowe et al. 2017), QMIX(Rashid et al. 2018), FOP(Zhang et al. 2021), and DOP(Wang et al. 2020), our proposed solution can be readily integrated into these algorithms by sampling the joint action-values. To this end, we proposed RGMComm, which trains an online clustering network via a Regularized Information Maximization (RIM) loss function (Hu et al. 2017), consisting of **a novel cosine-distance clustering loss** I_{CD} specifically designed to minimize the proven return bound for message label generation, and a mutual information regularization loss $I_{\mathrm{MI}}(o_j, m_j)$ to guarantee even cluster sizes and distinct observations-to-cluster assignments. The algorithm generates naturally interpretable discrete messages and easily extends to MARL problems with continuous state space by sampling the replay buffer. Evaluations using continuous state space problems (Lowe et al. 2017) show significant improvement over state-of-the-art communication-enabled MARL algorithms with partial observability, including CommNet(Sukhbaatar, Szlam, and Fergus 2016), MADDPG(Lowe et al. 2017), IC3Net (Singh, Jain, and Sukhbaatar 2018), TarMAC(Das et al. 2018), SARNet (Rangwala and Williams 2020), and VQ-VIB (Tucker et al. 2022).

The main contributions of the paper are as follows:

- We quantify the return gap between an ideal policy with full observability and a partially observable policy with communication via a closed-form upper-bound.
- We introduce RGMComm, a discrete communication framework for MARL, aimed at minimizing the upper bound of the return gap. Instead of generating messages directly via DNNs, RGMComm employs a finite-size discrete alphabet and treats message generation as an online clustering problem.
- With few-bit messages, RGMComm significantly outperforms state-of-the-art communication-enabled MARL algorithms with partial observability and achieves nearly-optimal returns.

Related Work

Previous work in MARL communication mostly establishes how agents should learn to communicate assuming continuous communication vectors (Foerster et al. 2016; Singh, Jain, and Sukhbaatar 2018; Jiang and Lu 2018; Das et al. 2018; Rangwala and Williams 2020). Inspired by the efficient communication patterns observed in human interactions, where people only use a discrete set of words, some previous works enable agents to learn sparse and discrete communication. CommNet (Sukhbaatar, Szlam, and Fergus 2016) and MADDPG (Lowe et al. 2017) learn continuous communication vectors alongside their policy and can generate discrete 1-hot binary communication vectors using a pre-defined set of communication alphabets. However, CommNet uses a large single network for all agents, so it cannot easily scale and would perform poorly in an environment with a large number of agents. MADDPG adapts the actor-critic framework using a centralized critic that takes as input the observations and actions of all agents and trains an independent policy network for each agent to generate communication as part of the actions, where each agent would learn a policy specializing in specific tasks. The policy network easily overfits the number of agents, which is infeasible in large-scale MARL.

Further, communication in terms of sequences of discrete symbols are investigated in (Havrylov and Titov 2017; Mordatch and Abbeel 2017). Both of these works generate categorical messages and adapt the Gumbel-Softmax Estimator (Jang, Gu, and Poole 2016; Maddison, Mnih, and Teh 2016) to make the communication models with discrete labels differentiable, training them with backpropagation algorithms. However, (Havrylov and Titov 2017) only studies a two-agent setting, and (Mordatch and Abbeel 2017) only learns message meanings through the prompt rewards in a small action space. Many works, therefore, resort to differentiable discrete communication such as (Freed et al. 2020; Lazaridou and Baroni 2020; Li et al. 2022; Tucker et al. 2022) where agents are allowed to directly optimize each other's communication policies through gradients. However, these approaches impose a strong constraint on the nature of communication, which limits their applicability to many real-world multi-agent coordination tasks. Besides, the primary limitation is that these approaches lack rigorous policy regret minimization guarantees, operating as heuristic designs.

Preliminaries and Problem Formulation

Dec-POMDP: A Dec-POMDP (Bernstein et al. 2002; Mei, Zhou, and Lan 2023) models cooperative MARL, where agents lack complete information about the environment and have only local observations. We formulate a Dec-POMDP with communication as a tuple $D = \langle S, A, P, \Omega, O, I, n, R, \gamma, g \rangle$, where S is the joint **state** space and $A = A_1 \times A_2 \times \cdots \times A_n$ is the joint **action** space, where $\boldsymbol{a} = (a_1, a_2, \ldots, a_n) \in A$ denotes the joint action of all agents. $P(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) : S \times A \times S \to [0, 1]$ is the **state transition**

function. Ω is the **observation** space. $O(\boldsymbol{s}, i) : S \times I \to \Omega$ is a function that maps from the joint state space to distributions of observations for each agent i. $I = \{1, 2, \ldots, n\}$ is a set of n agents, $R(\boldsymbol{s}, \boldsymbol{a}) : S \times A \to \mathbb{R}$ is the **reward function** in terms of state **s** and joint action $\boldsymbol{a}$, γ is the discount factor, and $g : \Omega \to M$ is the **message generation function** that each agent j uses to encode its local observation o_j into a communication message for other agents $i \neq j$. We use $\boldsymbol{m_{-i}} = \{m_j = g(o_j), \forall j \neq i\}$ to denote the collection of messages agent i receives from all other agents $j \neq i$. In Dec-POMDP with communications, each agent i considers an individual policy $\pi_i(a_i|o_i, \boldsymbol{m_{-i}})$ conditioned on local observation o_i and messages $\boldsymbol{m_{-i}}$, i.e., $\pi = [\pi_i(a_i|o_i, \boldsymbol{m_{-i}}), \forall i]$. To model the return gap, we also define an ideal policy with full observability: $\pi^* = [\pi_i^*(a_i|o_1, \ldots, o_n), \forall i]$.

Return Gap Minimization: Given a policy π, we consider the average expected return $J(\pi) = \lim_{T\to\infty}(1/T)E_\pi[\sum_{t=0}^{T} R_t]$. The goal of this paper is to minimize the return gap between an ideal policy $\pi^* = [\pi_i^*(a_i|o_1, \ldots, o_n), \forall i]$ with full observability and a partially-observable policy with communications $\pi = [\pi_i(a_i|o_i, \boldsymbol{m_{-i}}), \forall i]$ where message labels $\boldsymbol{m_{-i}} = \{m_j = g(o_j), \forall j \neq i\}$, i.e.,

$$\min J(\pi^*) - J(\pi, g). \tag{1}$$

While the problem is equivalent to maximizing $J(\pi, g)$, the return gap can be analyzed more easily by contrasting π and π^*. We derive an upper bound of the return gap and then design efficient communication strategies to minimize it. We consider the discounted observation-based state value and the corresponding action-value functions for the Dec-POMDP:

$$\begin{aligned} V^\pi(\boldsymbol{o}) &= \mathbb{E}_\pi[\sum_{i=0}^{\infty} \gamma^i \cdot R_{t+i} \Big| \boldsymbol{o}_t = \boldsymbol{o}, \boldsymbol{a}_t \sim \pi], \\ Q^\pi(\boldsymbol{o}, \boldsymbol{a}) &= \mathbb{E}_\pi[\sum_{i=0}^{\infty} \gamma^i \cdot R_{t+i} \Big| \boldsymbol{o}_t = \boldsymbol{o}, \boldsymbol{a}_t = \boldsymbol{a}], \end{aligned} \tag{2}$$

where t is the current time step. Re-writing the average expected return as an expectation in terms of $V^\pi(\boldsymbol{o})$:

$$J(\pi) = \lim_{\gamma\to 1} E_\mu[(1-\gamma)V^\pi(\boldsymbol{o})], \tag{3}$$

where μ is the initial observation distribution at time step $t = 0$, i.e., $\boldsymbol{o}(0) \sim \mu$. We will leverage this state-value function $V^\pi(\boldsymbol{o})$ and its corresponding action-value function $Q^\pi(\boldsymbol{o}, \boldsymbol{a})$ to unroll the Dec-POMDP and derive a closed-form upper-bound to quantify the return gap. For $V^\pi(\boldsymbol{o})$ and $Q^\pi(\boldsymbol{o}, \boldsymbol{a})$ we suppress g for simpler notations.

Upper-Bounding the Return Gap

Since directly minimizing the return gap is intractable, we prove that the return gap between ideal policy π^* with full observability and optimal communication-enabled policy π with partial observability is bounded by $O(\sqrt{\epsilon} n Q_{\max})$ with respect to average cosine-distance ϵ in each cluster and maximum action-value $Q_{\max}$ in Thm. 3.

In this section, we present the theoretical proof of Thm. 3 followed by an illustrative example. The proof consists of the following major steps: (i). We characterize the change in optimal expected average return $J(\pi^*) - J(\pi, g)$ for any two policies π^* (under full observability) and π (under partial observability with communication) based on the joint action-values under π^*. The result is stated as a Policy Change Lemma 1. (ii). We apply Policy Change Lemma 1 to two auxiliary policies $\pi^*_{(j)}$ and $\pi_{(j)}$, where $\pi^*_{(j)}$ is the optimal policy conditioned on complete observations $\boldsymbol{o} = [o_1, \ldots, o_j, \ldots, o_n]$ including observation o_j of agent j, and $\pi_{(j)}$ is the policy conditioned on partial observations $\boldsymbol{o} = [o_1, \ldots, o_{j_1}, m_j, o_{j+1} \ldots, o_n]$ without knowing observation o_j but including a communication message label $m_j = g(o_j)$ generated by agent j. The result (formulated in Lemma 2) allows us to quantify the impact of message-encoding agent j's local observations o_j as m_j, rather than having complete access to o_j, on the optimal expected average return gap. (iii). Finally, we construct a sequence of policies beginning from π^*, end with π, which has a process of changing by replacing each observation o_j by label m_j, one at a time: $\pi^* = [\pi_i(a_i|o_1, o_2, \ldots, o_n), \forall i] \to [\pi_i(a_i|m_1, o_2, \ldots, o_n), \forall i] \to [\pi_i(a_i|m_1, m_2, \ldots, o_n), \forall i] \to \ldots \to \pi$=$[\pi_i(a_i|m_1, \ldots, o_i, \ldots, m_n), \forall i]$. Applying the results from step (ii) (Lemma 2) for n times, we can quantify the desired return gap between π^* with full observability and communication-enabled policy π in Thm. 3. For simplicity, we use V^* to represent V^{π^*}, and Q^* to represent Q^{π^*}. We only give a sketch below, and **Appendix A (Chen, Lan, and Joe-Wong 2023) contains a notation table, and complete proofs for all theoretical results**.

We assume that observation/action spaces defined in the Dec-POMDP tuple are discrete with finite observations and actions, i.e., $|\Omega| < \infty$ and $|A| < \infty$. This is a technical condition to simplify the proof. For Dec-POMDPs with continuous observation and action spaces, the results can be easily extended by considering cosine-distance between action-value functions and replacing summations with integrals, or sampling the action-value functions as an approximation.

Lemma 1. *(Policy Change Lemma.) For any policies π^* and π, the optimal expected average return gap is bounded by:*

$$\begin{aligned} J(\pi^*) - J(\pi) &\leq \sum_m \sum_{\boldsymbol{o}\sim m} [Q^*(\boldsymbol{o}, \boldsymbol{a}_t^{\pi^*}) - Q^\pi(\boldsymbol{o}, \boldsymbol{a}_t^\pi)] d_\mu^\pi(\boldsymbol{o}), \\ d_\mu^\pi(\boldsymbol{o}) &= (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \cdot P(\boldsymbol{o}_t = \boldsymbol{o}|\pi, \mu), \end{aligned} \tag{4}$$

where $d_\mu^\pi(\boldsymbol{o})$ is the γ-discounted visitation probability under policy π and initial observation distribution μ, and $\sum_{\boldsymbol{o}\sim m}$ is a sum over all observations corresponding to message m.

Proof Sketch. Our key idea is to leverage the state value function $V^\pi(\boldsymbol{o})$ and its corresponding action value function $Q^\pi(\boldsymbol{o}, \boldsymbol{a})$ in Eq.(2) to unroll the Dec-POMDP from timestep $t = 0$ and onward. The detailed proof is provided in Appendix A (Chen, Lan, and Joe-Wong 2023).

Then we define the action value vector corresponding to observation o_j, i.e.,

$$\bar{Q}^*(o_j) = [\widetilde{Q}^*(o_{-j}, o_j), \forall o_{-j}], \tag{5}$$

where o_{-j} are the observations of all other agents and $\widetilde{Q}^*(o_{-j}, o_j)$ is a vector of action-values weighted by marginalized visitation probabilities $d^\pi_\mu(o_i|o_j)$ and corresponding to different actions, i.e., $\widetilde{Q}^*(o_{-j}, o_j) = [Q^*(o_{-j}, o_j, \boldsymbol{a}) \cdot d^\pi_\mu(o_{-j}|o_j), \forall \boldsymbol{a}]$.

Next, we view the message generation $m_j = g(o_j)$ as a clustering problem with o_j as input and m_j as labels, i.e., the o_j are divided into clusters and each is labeled with the clustering label to be sent out to all other agents $i \neq j$. We bound the policy gap between $\pi^*_{(j)}$ and $\pi_{(j)}$, which are optimal policies conditioned on o_j and m_j, using the average cosine-distance of action value vectors $\bar{Q}^*(o_j)$ corresponding to o_j in the same cluster and its cluster center $\bar{H}(m) = \sum_{o_j \sim m} \bar{d}_m(o_j) \cdot \bar{Q}^*(o_j)$ under each message m. Here $\bar{d}_m(o_j) = d^\pi_\mu(o_j)/d^\pi_\mu(m)$ is the marginalized probability of o_j in cluster m and $d^\pi_\mu(m)$ is the probability of message m under policy π, and the environments' initial observation distribution is represented by $\boldsymbol{o}(t=0) \sim \mu$. To this end, we let $\epsilon(o_j) = D_{cos}(\bar{Q}^*(o_j), \bar{H}(m))$ be the cosine-distance between vectors $\bar{Q}^*(o_j)$ and $\bar{H}(m)$ and consider the **average cosine-distance** ϵ across all clusters represented by different message labels m, which is defined as:

$$\epsilon \triangleq \sum_m d^\pi_\mu(m) \sum_{o_j \sim m} \bar{d}_m(o_j) \cdot \epsilon(o_j), \tag{6}$$

The result is summarized in Lemma 2.

Lemma 2. *(Impact of Communication.) Consider two optimal policies $\pi^*_{(j)}$ and $\pi_{(j)}$ conditioned on o_j and m_j, respectively, while the observability and policies of all other agents remain the same. The optimal expected average return gap is bounded by:*

$$J(\pi^*_{(j)}) - J(\pi_{(j)}, g) \leq O(\sqrt{\epsilon} Q_{\max}) \tag{7}$$

where $Q_{\max}$ is the maximum absolute action value of $\bar{Q}^*(o_j)$ in each cluster as $Q_{\max} = max_{o_j} ||\bar{Q}^*(o_j)||_2$, and ϵ is the average cosine-distance defined in Eq.(6).

Proof Sketch. We give an outline and provide the complete proof in Appendix A (Chen, Lan, and Joe-Wong 2023). Since the observability of all other agents $i \neq j$ remains the same, we consider them as a conceptual agent denoted by $-j$. For simplicity, we use π^* to represent $\pi^*_{(j)}$, and π to represent $\pi_{(j)}$ in distribution functions. The proof contains the following major steps: (i). Viewing the problem as a clustering of o_j, restrict policy $\pi_{(j)}$ (conditioned on m_j) to taking the same actions for all o_j in the same cluster under the same message m_j. (ii). Re-writing the optimal expected average return gap derived in Policy Change Lemma 1 in vector terms using action-value vectors $\bar{Q}^*(o_j)$ and an auxiliary maximization function $\Phi_{\max}(\bar{Q}^*(o_j))$. (iii). Projecting action-value vectors towards cluster centers to quantify the return gap through orthogonal parts related to projection errors. (iv). Deriving an upper bound on the return gap by bounding the orthogonal projection errors using the average cosine distance within each cluster.

Step 1: Recasting communication into online clustering. In this view, policy $\pi_{(j)}$ (conditioned on m_j) is restricted to taking the same actions for all o_j in the same cluster and under the same message m_j (the clustering label).

Step 2: Rewrite the return gap in vector form. We define an auxiliary function $\Phi_{\max}(\boldsymbol{X})$ that returns the largest component of vector $\boldsymbol{X}$. Since the optimal average return $J(\pi^*_{(j)})$ is conditioned on complete o_j, it can achieve the maximum for each vector $\bar{Q}^*(o_j)$. Thus, $\sum_{o_j \sim m} \bar{d}_m(o_j) \cdot \Phi_{\max}(\bar{Q}^*(o_j))$ can be defined as selecting the action from optimal policy $\pi^*_{(j)}$ where each agent chooses a different action distribution to maximize $(\bar{Q}^*(o_j))$. On the other hand, policy $\pi_{(j)}$ is conditioned on messages m_j rather than complete o_j and thus must take the same actions for all o_j in the same cluster. Hence we can construct a (potentially sub-optimal) policy to achieve $\Phi_{\max}(\sum_{o_j \sim m} \bar{d}_m(o_j) \cdot \bar{Q}^*(o_j))$ which provides a lower bound on $J(\pi_{(j)}, g)$. (The Appendix A (Chen, Lan, and Joe-Wong 2023) details the transformation to vector terms with the help of $\Phi_{\max}(\boldsymbol{X})$.)

Using the transformed vector term formats of optimal returns $J(\pi^*_{(j)})$ and $J(\pi_{(j)}, g)$ ($J(\pi_{(j)}, g)$ is lower bounded), we can obtain an upper bound on the return gap:

$$\begin{aligned} & J(\pi^*_{(j)}) - J(\pi_{(j)}) \\ & \leq \sum_m d^\pi_\mu(m) [\sum_{o_j \sim m} \bar{d}_m(o_j) \cdot \Phi_{\max}(\bar{Q}^*(o_j)) \\ & - \Phi_{\max}(\sum_{o_j \sim m} \bar{d}_m(o_j) \cdot \bar{Q}^*(o_j))], \end{aligned} \tag{8}$$

Step 3: Projecting action-value vectors toward cluster centers. The policy $\pi_{(j)}$ conditioned on m_j takes actions based on the action-value vector at the cluster center $\bar{H}(m)$. For each pair of two vectors $\bar{Q}^*(o_j)$ and $\bar{H}(m)$ with $D(\bar{Q}^*(o_j), \bar{H}(m)) \leq \epsilon(o_j)$, we use $\cos\theta_{o_j}$ to denote the cosine-similarity between each $\bar{Q}^*(o_j)$ and its center $\bar{H}(m)$. Then we have the cosine distance $D(\bar{Q}^*(o_j), \bar{H}(m)) = 1 - \cos\theta_{o_j} \leq \epsilon(o_j)$. By projecting $\bar{Q}^*(o_j)$ toward $\bar{H}(m)$, $\bar{Q}^*(o_j)$ could be re-written as $\bar{Q}^*(o_j) = Q^\perp(o_j) + \cos\theta_{o_j} \cdot \bar{H}_m$, then we could upper bound $\Phi_{max}(\bar{Q}^*(o_j))$ by:

$$\Phi_{\max}(\bar{Q}^*(o_j)) \leq \Phi_{\max}(\cos\theta_{o_j} \cdot \bar{H}_m) + \Phi_{\max}(Q^\perp(o_j)).$$

Taking a sum over all o_j in the cluster, we have $\sum_{o_j \sim m} \bar{d}_m(o_j) \Phi_{\max}(\cos\theta_{o_j} \cdot \bar{H}_m) = \Phi_{\max}(\bar{H}_m)$, since the projected components $\cos\theta_{o_j} \cdot \bar{H}_m$ should add up to exactly $\bar{H}_m$. To bound Eq.(8)'s return gap, it remains to bound the orthogonal components $Q^\perp(o_j)$.

Step 4: Deriving the upper bound w.r.t. cosine-distance. Let $||\cdot||_2$ be the L_2 norm, then the maximum function $\Phi_{\max}(Q^\perp(o_j))$ can be bounded by its L_2 norm $C \cdot ||Q^\perp(o_j)||_2$ for some constant C, i.e., $\Phi_{max}(Q^\perp(o_j)) \leq C \cdot ||Q^\perp(o_j)||_2$. Since $Q^\perp(o_j) = \bar{Q}^*(o_j) \cdot sin(\theta)$, $\Phi_{\max}(Q^\perp(o_j))$ can be further upper bounded by $C \cdot ||\bar{Q}^*(o_j)||_2 \cdot |\sin\theta|$. Denote a constant $Q_{\max}$ as the maximum absolute action value of $\bar{Q}^*(o_j)$ in each cluster as $Q_{\max} = max_{o_j} ||\bar{Q}^*(o_j)||_2$; combined with $|sin(\theta)| = \sqrt{1 - cos^2(\theta)} = \sqrt{1 - [1 - \epsilon(o_j)]^2}$, we can thus obtain

$$\Phi_{max}(Q^\perp(o_j)) \leq O(\sqrt{\epsilon(o_j)} Q_{\max}). \tag{9}$$

Using the concavity of the square root with Eq.(6), i.e., $\sum_m d_\mu^\pi(m)[\sum_{o_j\sim m} \bar{d}_m(o_j)\cdot\sqrt{\epsilon(o_j)}] \le \sqrt{\epsilon}$, we derive the desired upper bound $J(\pi^*_{(j)}) - J(\pi_{(j)}) \le O(\sqrt{\epsilon}Q_{\max})$.

Theorem 3. *In n-agent Dec-POMDP, the return gap between policy π^* with full-observability and communication-enabled policy π with partial-observability is bounded by:*

$$J(\pi^*) - J(\pi, g) \le O\left(\sqrt{\epsilon} n Q_{\max}\right). \tag{10}$$

Beginning from $\pi^* = [\pi_i^*(a_i|o_1, o_2, \ldots, o_n), \forall i]$, we can construct a sequence of n policies, each replacing the conditioning on o_j by messages m_j, for $j = 1$ to $j = n$, one at a time. This will result in policy π with partial observability. Applying Lemma 2 for n times, we prove the upper bound between $J(\pi^*)$ and $J(\pi, g)$ in this theorem.

Remark 4. Thm. 3 holds for any arbitrary finite number of message labels $|M|$. Furthermore, increasing $|M|$ reduces the average cosine distance (since more clusters are formed) and, consequently, a reduction in the return gap due to the upper bound derived in Thm. 3.

An illustrative example. We consider a two-agent matrix game with pay-off Q^* shown in Fig 1. Assume that different observations are equally likely and that agent 2 has a single action in each state. It is easy to see that under an ideal optimal policy $\pi^* = [\pi_1^*, \pi_2^*]$ with full observability (o_1, o_2) can achieve an optimal average reward of $J(\pi_1^*) = 1/2\cdot[(53.2+4.5+64.0+16.1)/4+(31.8+28.0+34.1+81.5)/4] = 39.1375$ by choosing the optimal actions in each (o_1, o_2), i.e., $\pi^* = \text{argmax}_{a_1,a_2} Q^*(o_1, o_2, a_1, a_2)$. Consider a POMDP scenario where agent 1 has only local observation o_1, and agent 2 can send a 1-bit message $m_2 = g(o_2)$ encoding its local observation o_2. Since the message is only 1-bit but there are 4 possible observations o_2, multiple observations must be encoded into the same message. Thus, agent 1 is restricted to a limited policy class $\pi_1(a_1|o_1, m_2)$ taking the same actions (or action distributions) under the same o_1 and for all o_2 corresponding to the same message m_2. Thm. 3 shows that to minimize the return gap between $\pi_1(a_1|o_1, m_2)$ with partial observation and the ideal optimal policy $\pi^* = [\pi_1^*, \pi_2^*]$ with full observability, we can leverage a message generation function to minimize the average cosine distance ϵ between action values for different observations o_2. In particular, let $\bar{Q}^*(o_2)$ be a column of Q^* corresponding to o_2. There are then four possibilities of o_2: $o_{21}, o_{22}, o_{23}, o_{24}$. If $\bar{Q}^*(o_{2i})$ and $\bar{Q}^*(o_{2j})$ are likely maximized at the same actions a_1, then encoding

Agent 1 \ Agent 2		o_{21}	o_{22}	o_{23}	o_{24}
		a_{21}^*	a_{22}^*	a_{23}^*	a_{24}^*
o_{11}	a_{11}	53.2	4.5	58.5	0.3
	a_{12}	42.9	1.2	64.0	16.1
o_{12}	a_{11}	31.8	2.0	34.1	6.8
	a_{12}	22.9	28.0	10.0	81.5
		$\bar{Q}^*_{(o_{21})}$	$\bar{Q}^*_{(o_{22})}$	$\bar{Q}^*_{(o_{23})}$	$\bar{Q}^*_{(o_{24})}$

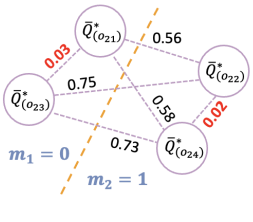


Figure 1: An illustrative example of optimizing message generation m_2 via clustering to minimize the return gap.

o_{2i} and o_{2j} to the same message would result in little return gap, as formalized in Thm. 3. The message generation could be viewed as a clustering problem over o_2 under vector cosine-distance (Muflikhah and Baharudin 2009). The right figure in Fig 1 displays the cosine distances calculated between every pair of action value vectors in $\bar{Q}^*(o_{2i})$ and $\bar{Q}^*(o_{2j})$ (which are readily available during centralized training). According to Thm. 3, assigning the same message labels to observations o_2 whose optimal action value vectors have smaller cosine distances would result in smaller return gaps. Hence, we assign message label '0' to $\bar{Q}^*(o_{21})$ and $\bar{Q}^*(o_{23})$, whose cosine distance is 0.03, and message label '1' to $\bar{Q}^*(o_{22})$ and $\bar{Q}^*(o_{24})$, whose cosine distance is 0.02. Accordingly, the message generation in Fig 1 leads to an average return of $J(\pi, g) = 38.775$ with an optimal return gap of $J(\pi^*) - J(\pi, g) = 0.3625$.

RGMComm Design: Minimizing the Upper Bound of Average Cosine Distance ϵ

The result in Thm. 3 inspires a new MARL communication framework - RGMComm shown in Fig.(2), which minimizes the return gap between fully-observable π^* and communication-enabled π by training an online clustering network g for message label generation, minimizing ϵ with a cosine-distance loss. The method can be easily implemented since:

- Due to the actor-critic framework adopted by most MARL algorithms, e.g., MADDPG(Lowe et al. 2017), QMIX(Rashid et al. 2018), FOP(Zhang et al. 2021), and DOP(Wang et al. 2020), joint action-value functions (often represented using DNNs) are readily available for training the message generation functions.
- To train communication (using online clustering) together with agent policies, we can efficiently sample action-values based on transitions and state visitations in the replay buffer, allowing easy application to continuous state-space problems.

MARL training: We can leverage state-of-the-art MARL algorithms adopting the actor-critic framework to train agent policies and obtain an estimate of the action-value function. Note that we drop time t in the following notations for simplicity. Let $\hat{Q}_\omega(\boldsymbol{o}, \boldsymbol{a})$ be a DNN parameterized by ω. We

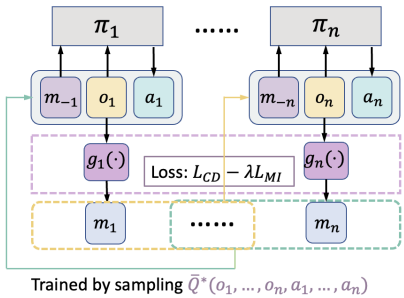


Figure 2: RGMComm trains message generation function g via sampled action-value vectors, shaping the cosine-distance loss function L_{CD}.

update the parameter ω using an experience replay buffer $\mathcal{R}$ containing tuples $(\boldsymbol{o}, \boldsymbol{a}, R, \boldsymbol{o}')$ and recording the experiences of all agents. The action-value function $\hat{Q}_\omega(\boldsymbol{o}, \boldsymbol{a})$ is updated by minimizing the loss function $\mathcal{L}(\omega)$ defined as:

$$\begin{aligned} \mathcal{L}(\omega) &= E_{(\boldsymbol{o},\boldsymbol{a},R,\boldsymbol{o}')}[(\hat{Q}_\omega(\boldsymbol{o}, \boldsymbol{a}) - y)^2], \\ y &= R + \gamma \hat{Q}_{\omega'}(\boldsymbol{o}', \boldsymbol{a}')|_{\boldsymbol{a}'} = \pi'(\boldsymbol{o}'). \end{aligned} \tag{11}$$

We note that decisions above may be taken with respect to decentralized policies conditioned on partial observation and communication messages, denoted as $\pi = [\pi_i(a_i|o_i, \boldsymbol{m_{-i}})|_{\boldsymbol{m_{-i}}=\{g(o_j),\forall j\neq i\}}, \forall i]$. Suppose that agents' policies are parameterized by $\{\theta_1, \ldots, \theta_n\}$, respectively. These policies are updated and executed in a decentralized manner using the policy gradient to minimize the expected return $J(\theta_i) = E[R_i]$ of each agent i:

$$\nabla_{\theta_i} J(\theta_i) = E_{\boldsymbol{o},\boldsymbol{a}\sim\mathcal{R}}\left[\nabla_{\theta_i} \log \pi_{\theta_i}(a_i|o_i, \boldsymbol{m_{-i}}) \hat{Q}_\omega^{\pi_{\theta_i}}(\boldsymbol{o}, \boldsymbol{a})\right]. \tag{12}$$

Learning message generation: The message generation functions $g = \{g_1, \ldots, g_n\}$ of all agents are approximated using DNNs parameterized by $\xi = \{\xi_1, \ldots, \xi_n\}$. For each agent j, we first sample a random minibatch of K_1 samples $\mathcal{X}_j = (\boldsymbol{o}^{k_1}, \boldsymbol{a}^{k_1}, R^{k_1}, \boldsymbol{o}'^{k_1})$ from the transitions recorded in replay buffer $\mathcal{R}$, which contains the observation-action pairs from all agents including agent j. Then we sample a set $\mathcal{X}_{-j} = (\boldsymbol{o}_{-j}^{k_2}, \boldsymbol{a}_{-j}^{k_2})$ from $\mathcal{X}_j$, which are the top K_2 frequent observation-action pairs in the minibatch $\mathcal{X}_j$ after removing o_j and a_j from $\mathcal{X}_j$. Then we form the sampled trajectories by combining (o_j, a_j) in $\mathcal{X}_j$ and $(\boldsymbol{o}_{-j}, \boldsymbol{a}_{-j})$ in $\mathcal{X}_{-j}$ as $\mathcal{D} = (\boldsymbol{o}^{k_1 k_2}, \boldsymbol{a}^{k_1 k_2}, R^{k_1 k_2}, \boldsymbol{o}'^{k_1 k_2})$. To obtain the action-values for clustering, we query the critic networks with $\mathcal{D}$ as the input to get the $\hat{Q}_\omega(o_j, \boldsymbol{o}_{-j}, a_j, \boldsymbol{a}_{-j})$, which approximates action-value vectors $\bar{Q}^*(o_j)$ defined in Eq. (5) and the illustrative example. We use $\bar{Q}^*(o_j)$ instead of $\hat{Q}_\omega$ in the following part to be consistent with the theoretical results. The message $m_j = g_{\xi_j}(o_j)$ is updated by minimizing a Regularized Information Maximization (RIM) loss function (Hu et al. 2017) $\mathcal{L}(g_{\xi_j})$ in terms of $\bar{Q}^*(o_j)$:

$$\begin{aligned} &\mathcal{L}(g_{\xi_i}) = L_{CD} - \lambda L_{MI}, \\ &L_{CD} = \sum_{p=1}^{K_1} \sum_{q\in N_{K_3}(p)} \left[D_{cos}(\bar{Q}^*(o_j^p), \bar{Q}^*(o_j^q)\right] \|m_j^p - m_j^q\|^2, \\ &L_{MI} = I(o_j; m_j) = H(m_j) - H(m_j|o_j), \end{aligned} \tag{13}$$

where L_{CD} is a clustering loss in the form of Locality-preserving loss (Huang et al. 2014). It preserves the locality of the clusters by pushing nearby data points of action-value vectors together. Inside L_{CD}, $o_j^p \in \mathcal{X}_j, p = 1, \ldots, K_1$ is the sampled observation o_j, $N_{K_3}(p)$ is the set of the K_3 nearest neighbors of $\bar{Q}^*(o_j^p)$, with D_{cos} (the cosine-distance between $\bar{Q}^*(o_j^p)$ and its neighbor $\bar{Q}^*(o_j^q)$) as the metric to define the neighbors. The mutual information loss L_{MI} measures the mutual information between observation o_j and message m_j. Here we measure the mutual information loss as the difference between the marginal entropy $H(m_j)$ and conditional entropy $H(m_j|o_j)$, which are defined as:

$$\begin{aligned} H(m_j) &= h(p_{\xi_j}(m_j)) = h(1/K_1) \sum_{p=1}^{K_1} p_{\xi_j}(m_j|o_j), \\ H(m_j|o_j) &= 1/K_1) \sum_{p=1}^{K_1} h(p_{\xi_j}(m_j|o_j). \end{aligned} \tag{14}$$

where $h(p(x)) = -\sum_{x'} p(x') \log p(x')$ is the entropy function. Increasing the marginal entropy $H(m_j)$ ensures diverse and uniformly-sized clusters, while decreasing the conditional entropy $H(m_j|o_j)$ leads to clear and definitive cluster assignments for each observation and thus minimizes overlap. In summary, minimizing L_{CD} narrows the return gap between full-observability and partial-observability policies with communication, and maximizing L_{MI} guarantees even cluster sizes and distinct observations-to-cluster assignments. A hyper-parameter λ balances these objectives.

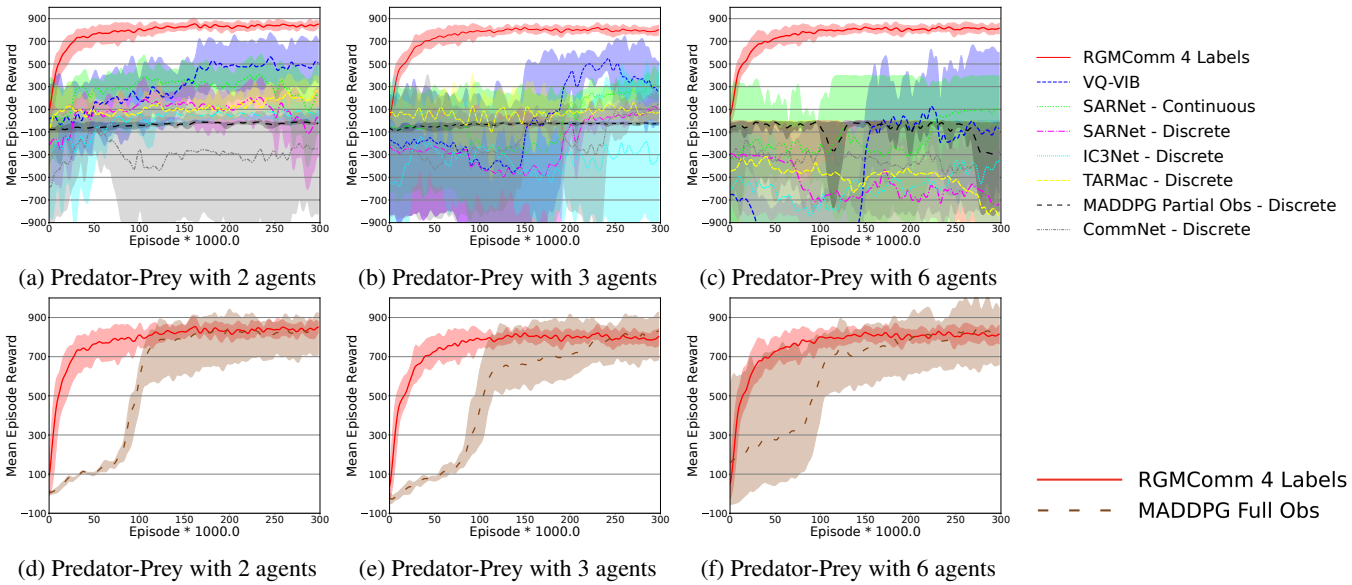
Action-value vectors are normalized and processed through the activation function for improved clustering outcomes. The choice of activation function is discussed in the experiments section. The Return-Gap-Minimization Communication(RGMComm) message label generation process is summarized in Algorithm 1, The complete pseudo-code for training RGMComm is in Appendix B (Chen, Lan, and Joe-Wong 2023).

Algorithm 1: RGMComm message label generation

Input: K_1, K_2, K_3, λ, Replay buffer $\mathcal{R}$, current parameters ω, $\xi = \{\xi_1, \ldots, \xi_n\}$.
for $t = 1$ **to** T **do**
  for agent j to n **do**
    Get top-K_1 samples $\mathcal{X}_j = (\boldsymbol{o}^{k_1}, \boldsymbol{a}^{k_1}, R^{k_1}, \boldsymbol{o}'^{k_1})$ from reply buffer $\mathcal{R}$;
    Get top-K_2 samples $\mathcal{X}_{-j} = (\boldsymbol{o}_{-j}^{k_2}, \boldsymbol{a}_{-j}^{k_2})$ from $\mathcal{X}_j$ from $\mathcal{X}_j$;
    combining (o_j, a_j) in $\mathcal{X}_j$ and $(\boldsymbol{o}_{-j}, \boldsymbol{a}_{-j})$ in $\mathcal{X}_{-j}$ as $\mathcal{D} = (\boldsymbol{o}^{k_1 k_2}, \boldsymbol{a}^{k_1 k_2}, R^{k_1 k_2}, \boldsymbol{o}'^{k_1 k_2})$;
    Query the critic ω with $\mathcal{D}$ as the input to get the $\hat{Q}_\omega(o_j, \boldsymbol{o}_{-j}, a_j, \boldsymbol{a}_{-j})$;
    Update g_{ξ_j} by minimizing the loss $L(g_{\xi_j})$ defined in the main paper;
  end for
end for
Output: Message functions: $m_j = g_{\xi_j}(o_j)$.

Experiments

We test RGMComm on continuous state space Dec-POMDP problems (using Multi-Agent Particle Environment (Lowe et al. 2017)). We compare it against strong baselines: **(1). Continuous communication** using SARNet (Rangwala and Williams 2020), which employs memory-based attention for uninterrupted messages. **(2).** One-hot **discrete communication** using CommNet (Sukhbaatar, Szlam, and Fergus 2016), MADDPG (Lowe et al. 2017), IC3Net (Singh, Jain, and Sukhbaatar 2018), TarMAC (Das et al. 2018), and SARNet (Rangwala and Williams 2020). We enable their discrete



(a) Predator-Prey with 2 agents (b) Predator-Prey with 3 agents (c) Predator-Prey with 6 agents

(d) Predator-Prey with 2 agents (e) Predator-Prey with 3 agents (f) Predator-Prey with 6 agents

Figure 3: Evaluation on n-RGMComm-trained predator agents in Predator-Prey tasks: (a)-(c) Comparing RGMComm with baselines with communication: RGMComm (red curve) converges to a higher mean episode reward than all the baselines. (d)-(f) Comparing RGMComm with full-observability policy: RGMComm (red curve) achieves nearly optimal mean episode reward (brown dashed curve) in all scenarios with varying numbers of agents, which illustrates its ability to minimize the return gap.

communication mode, where agents send discrete unstructured vectors. These baselines use Gumbel softmax for gradients and backpropagation. **(3). Vector-Quantized** Variational Information Bottleneck (VQ-VIB) (Tucker et al. 2022), the recent method using autoencoder and quantization for communication. All baselines train from scratch with the same hyperparameters as RGMComm, repeated 3 times with different seeds. Appendix C (Chen, Lan, and Joe-Wong 2023) has more details on baselines, architectures, hyperparameters, and environments.

Predator-Prey

In the Predator-Prey scenario, predators are trained to collaborate to surround and seize **prey who move randomly**. We trained RGMComm with different alphabet sizes $|M|$ and $N = 2, 3, 6$ **predators**. Figures 3a to 3c show that RGMComm outperforms all baselines with 2, 3, and 6 cooperative **predator** agents. The figures display learning curves of 300,000 episodes in terms of the mean episode reward, averaged over all evaluating episodes (10 episodes evaluated every 1000 episodes). Figures 3d to 3f show the learning curves of average returns under our RGMComm policy compared with an ideal full-observability policy π^*. RGMComm achieves near-optimal mean episode rewards in all scenarios with varying numbers of agents, which demonstrates its ability to minimize the return gap. Additionally, the message generation training process uses an online clustering algorithm that does not require a large volume of data, leading to faster convergence that scales well with more agents.

Fig 4 shows how the total number of message labels affects RGMComm's performance. We compare the normalized mean episode reward achieved by RGMComm with $|M| = 0, 2, 4, 8, 16, 32$ message labels. As expected, the mean episode reward increases with the number of message labels. Remarkably, with only $|M| = 2$ message labels (i.e., 1-bit communication), RGMComm achieves nearly optimal mean episode reward. With more than $|M| = 4$ message labels, RGMComm's reward exceeds that of the policy with full observability, this is because the message generation function learned from the critic provides a succinct, discrete representation of the optimal action-value structure, leading to less noisy communication signals and allowing agents to discover more efficient decision-making policies conditioned on the message labels.

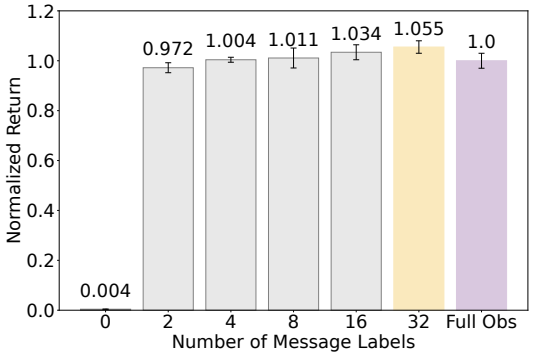
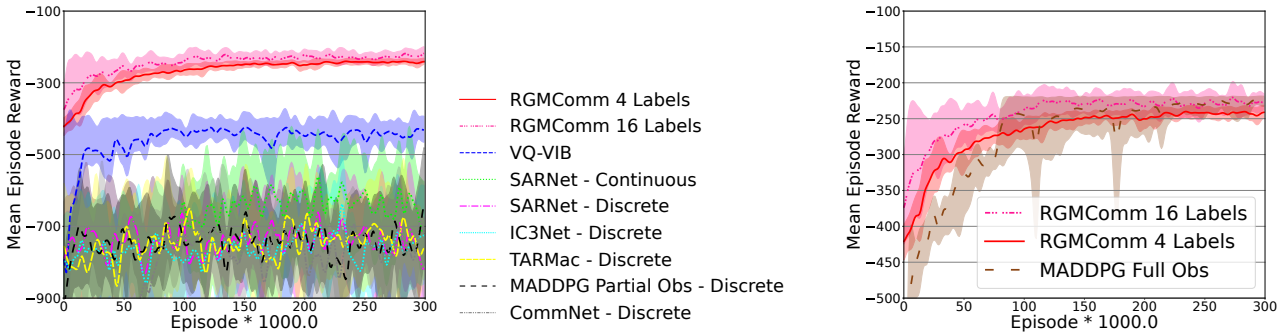


Figure 4: The normalized mean episode reward increases as the total number of message labels increases.



(a) RGMComm using 4 (red solid curve) and 16 message labels (pink dotted curve) both have higher convergence values than all baselines with partial observability - RGMComm via discrete communication allows more efficient policies to be learned in POMDP.

(b) RGMComm using 4 or 16 message labels both lead to almost zero return gap achieved by full observability policy (brown dashed curve).

Figure 5: Cooperation Navigation Experiments

Cooperative Navigation

We train RGMComm using 4 and 16 message labels for Cooperative Navigation, which tasks agents with collaboratively navigating to specific targets without collisions. Each agent could observe the nearest landmark with relative positions and velocities without knowing the others' information. At each timestep, the reward of an agent is given by the distance between the agent and its nearest landmark. A penalty will be added if a collision occurs between agents.

Fig. 5a shows the learning curve in terms of mean episode reward for RGMComm using 4 and 16 message labels compared with baselines with communication in partially observable environments. We can see that RGMComm converges to a much higher reward than all the baselines at a faster convergence speed. Figure 5b shows that RGMComm using 4 message labels leads to almost zero return gap and achieves nearly optimal returns. With 16 message labels, the proposed RGMComm algorithm sometimes even converges to a higher mean episode reward than the policy with full observability. The improved performance is due to the message generation function, learned from the centralized critic, offering a compact and distinct representation of the optimal action-value structure. This leads to clearer, less noisy communication signals, enabling agents to develop more effective decision-making strategies based on the message labels.

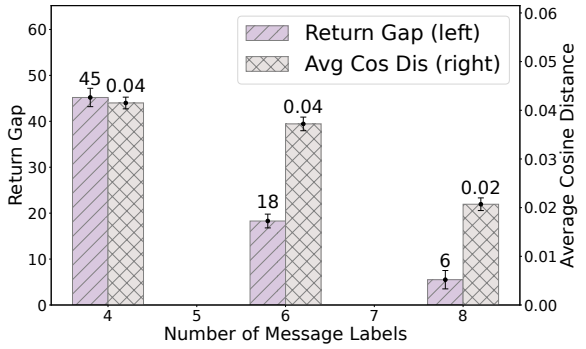


Figure 6: The return gap (left) is bounded by average cosine-distance (right) and diminishes as average cosine-distance (right) decreases due to the use of more message labels.

Figure 6 plots the average cosine-distance ϵ across clusters and the corresponding return gap (between RGMComm and the full-observability policies) using 4, 6, and 8 message labels. The cosine-distance ϵ is averaged over different clusters and different vectors $\bar{Q}^*(o_j)$ in each cluster. We note that the $Q_{\max}$ for 4-, 6-, 8-label RGMComm policy is $721, 739, 754$ respectively. To calculate the return gap, we evaluate RGMComm and full-observation policy with respect to the mean episode rewards by taking the average over the last 30 episodes after each algorithm converged. The numerical results justify Thm. 3's analysis: the return gaps are indeed bounded by $O(\sqrt{\epsilon} n Q_{\max})$, and the average return gap diminishes as the average cosine-distance over all clusters decreases due to using more message labels, which also validates the results in Remark 4.

Ablation Study

We further provide an ablation study on the RGMComm algorithm about how different normalizations of the action value vectors $\bar{Q}^*(o_j)$ (which result in different distance measures) affect the performance of RGMComm. We compare the achieved average returns by using the Softmax function, hyperbolic tangent (Tanh) function, and not using any activation functions in our proposed RGMComm algorithm with two baselines - independent Q learning and centralized Q learning algorithms (by allowing π^* to have full-observability). The results in Fig. 7 first show that RGMComm with Softmax (pink densely dashed curve) could achieve a higher return than RGMComm with no activation function (yellow densely dotted curve), and Independent Q learning algorithm (gray dash-dotted curve). This is because the Softmax activation function extracts the largest element of $\bar{Q}^*(o_j)$ for the distance calculation, this result validates the importance of cosine-distance in our bound. Fig 7 also shows that RGMComm with Tanh (purple solid curve) can converge to a higher return, compared to RGMComm with Softmax, this

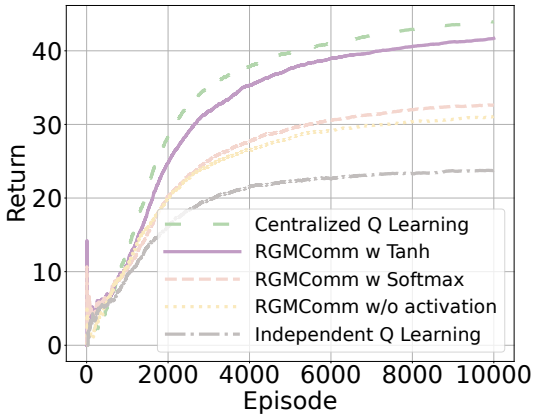


Figure 7: RGMComm with Tanh activation function (purple solid line) outperforms all baselines with other activation functions and converges to almost the optimal mean episode reward achieved by centralized Q learning (green loosely dashed line).

is because such S-shape functions like Tanh can push larger action-values toward 1, which helps the online clustering algorithm under cosine-distance to group the action-values that are likely maximized at the same actions. Fig 7 further shows that RGMComm with Tanh can achieve a nearly optimal return compared to the centralized Q learning full observability algorithm (green loosely dashed curve), indicating its ability to minimize the return gap as well.

Communication Message Interpretations

To illustrate message interpretation, we run RGMComm on a grid-world maze environment (see environments details in Appendix C (Chen, Lan, and Joe-Wong 2023)) for easy visualization, while our method here can be readily used in other environments. The discrete message labels used in RGMComm are naturally interpretable. We investigate the correlation between message labels and local observations as well as agent actions. We visualize the count of message labels agents uttered in different positions during training in Fig 8a and Fig 8b. We use 'red', 'yellow', 'purple', and 'grey' to represent the positions where agents have higher probabilities to send message labels '0', '1', '2', and '3', respectively. We can see that when agents are closer to the high-reward target(circle), they are more likely to send label '1', while more likely to send label '2' when they are closer to the lower-reward target(triangle). To interpret other messages, for two RGMComm agents, we also visualize the correlation between actions and messages they receive. As shown in Fig 8c and Fig 8d, both agents are more likely to take action 'down', 'up', 'right', 'left' conditioned receiving label '0', '1', '2', and '3', respectively. Putting these together, for instance, we can see that when an RGMComm agent approaches the high-reward target, it will utter a label '1', instructing other agents to move 'up' or 'left', which effectively guides other agents to approach the same target located at the upper left

(a) Label-position (agent 1)

(b) Label-position (agent 2)

(c) Label-action (agent 1)

(d) Label-action (agent 2)

Figure 8: (a)-(b): agents are more likely to send label '1' when closer to higher-reward target (circle), while more likely to send label '2' when closer to lower-reward target (triangle); (c)-(d): both agents are more likely to take action 'down', 'up', 'right', 'left' conditioned receiving label '0', '1', '2', and '3', respectively.

corner of the map. Discrete communication in RGMComm naturally allows such interpretations to be obtained during training in different task contexts.

Conclusion

In this paper, we propose a discrete communication framework for MARL using any finite-size discrete alphabet that views message generation as an online clustering problem and quantifies the optimal return gap between an ideal policy and a communication-enabled policy with a closed-form upper bound. We also show that the proposed new class of MARL communication algorithm which minimizes the return gap with few-bit messages significantly outperforms state-of-the-art baselines and achieves nearly-optimal returns. For future work, we will investigate the decision paths of agents through a more explainable structure than DNNs, and see how the decision paths affect each other in MARL settings.

Acknowledgements

This research was funded by multiple sources: Office of Naval Research (ONR) grants N00014-23-1-2532 and N00014-23-1-2850, a CISCO Research Award, and also in part by the Office of Naval Research under Grant W911NF19110036. The views, opinions, findings, conclusions, or recommendations presented in this material are solely those of the author(s) and may not necessarily represent the perspectives of the Office of Naval Research.

References

Bernstein, D. S.; Givan, R.; Immerman, N.; and Zilberstein, S. 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of operations research*, 27(4): 819–840.

Cao, Y.; Yu, W.; Ren, W.; and Chen, G. 2013. An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination. *IEEE Transactions on Industrial Informatics*, 9(1): 427–438.

Chen, J.; Lan, T.; and Choi, N. 2023. Distributional-Utility Actor-Critic for Network Slice Performance Guarantee. In *Proceedings of the Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 161–170.

Chen, J.; Lan, T.; and Joe-Wong, C. 2023. RGMComm: Return Gap Minimization via Discrete Communications in Multi-Agent Reinforcement Learning. arXiv:2308.03358.

Chen, J.; Wang, Y.; and Lan, T. 2021. Bringing fairness to actor-critic reinforcement learning for network utility optimization. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, 1–10. IEEE.

Chen, J.; Zhang, L.; Riem, J.; Adam, G.; Bastian, N. D.; and Lan, T. 2023. RIDE: Real-time Intrusion Detection via Explainable Machine Learning Implemented in a Memristor Hardware Architecture. In *2023 IEEE Conference on Dependable and Secure Computing (DSC)*, 1–8. IEEE.

Das, A.; Gervet, T.; Romoff, J.; Batra, D.; Parikh, D.; Rabbat, M.; and Pineau, J. 2018. TarMAC: Targeted Multi-Agent Communication.

Foerster, J. N.; Assael, Y. M.; de Freitas, N.; and Whiteson, S. 2016. Learning to Communicate with Deep Multi-Agent Reinforcement Learning.

Freed, B.; James, R.; Sartoretti, G.; and Choset, H. 2020. Sparse discrete communication learning for multi-agent cooperation through backpropagation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7993–7998. IEEE.

Havrylov, S.; and Titov, I. 2017. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. *Advances in neural information processing systems*, 30.

Hu, W.; Miyato, T.; Tokui, S.; Matsumoto, E.; and Sugiyama, M. 2017. Learning discrete representations via information maximizing self-augmented training. In *International conference on machine learning*, 1558–1567. PMLR.

Huang, P.; Huang, Y.; Wang, W.; and Wang, L. 2014. Deep Embedding Network for Clustering. In *2014 22nd International Conference on Pattern Recognition*, 1532–1537.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Jiang, J.; and Lu, Z. 2018. Learning Attentional Communication for Multi-Agent Cooperation.

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

Li, S.; Zhou, Y.; Allen, R.; and Kochenderfer, M. J. 2022. Learning Emergent Discrete Message Communication for Cooperative Reinforcement Learning. In *2022 International Conference on Robotics and Automation (ICRA)*, 5511–5517.

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

Maddison, C. J.; Mnih, A.; and Teh, Y. W. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Mei, Y.; Zhou, H.; and Lan, T. 2023. Remix: Regret minimization for monotonic value function factorization in multiagent reinforcement learning. *arXiv preprint arXiv:2302.05593*.

Mei, Y.; Zhou, H.; Lan, T.; Venkataramani, G.; and Wei, P. 2023. MAC-PO: Multi-agent experience replay via collective priority optimization. *arXiv preprint arXiv:2302.10418*.

Mordatch, I.; and Abbeel, P. 2017. Emergence of Grounded Compositional Language in Multi-Agent Populations.

Muflikhah, L.; and Baharudin, B. 2009. Document clustering using concept space and cosine similarity measurement. In *2009 International Conference on Computer Technology and Development*, volume 1, 58–62. IEEE.

Rangwala, M.; and Williams, R. 2020. Learning Multi-Agent Communication through Structured Attentive Reasoning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713829546.

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

Singh, A.; Jain, T.; and Sukhbaatar, S. 2018. Learning when to Communicate at Scale in Multiagent Cooperative and Competitive Tasks.

Sukhbaatar, S.; Szlam, A.; and Fergus, R. 2016. Learning Multiagent Communication with Backpropagation.

Tucker, M.; Levy, R.; Shah, J. A.; and Zaslavsky, N. 2022. Trading off utility, informativeness, and complexity in emergent communication. *Advances in neural information processing systems*, 35: 22214–22228.

Tucker, M.; Li, H.; Agrawal, S.; Hughes, D.; Sycara, K.; Lewis, M.; and Shah, J. A. 2021. Emergent discrete communication in semantic spaces. *Advances in Neural Information Processing Systems*, 34: 10574–10586.

Wang, T.; Wang, J.; Zheng, C.; and Zhang, C. 2019. Learning nearly decomposable value functions via communication minimization. *arXiv preprint arXiv:1910.05366*.

Wang, Y.; Han, B.; Wang, T.; Dong, H.; and Zhang, C. 2020. Dop: Off-policy multi-agent decomposed policy gradients. In *International Conference on Learning Representations*.

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