COMMUNICATION GAIN AND DELAY COST UNDER CROSS-TIMESTEP DELAYS IN COOPERATIVE MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Communication is crucial for cooperative multi-agent reinforcement learning (MARL) under partial observability, but bounded cross-timestep communication delays make received messages temporally misaligned and potentially stale. We formalize this setting as a delayed-communication partially observable Markov game (DeComm-POMG) and introduce Communication Gain and Delay Cost (CGDC), a gain–cost metric that evaluates delayed-message utility through no-message and timely-reference counterfactuals. We further derive a delay-induced return-loss bound that relates the return gap between policies using timely references and those using delayed messages to the accumulated information gap. Guided by CGDC, we propose CDCMA, an actor–critic framework for selecting communication partners, constructing future-aware outgoing messages, and aggregating received delayed messages. Experiments on MPE tasks without teammate vision and on SMAC maps show that CDCMA achieves higher mean performance than the communication baselines across the evaluated bounded stochastic delay regimes; on two representative tasks, it also yields smaller decision-distribution gaps. CDCMA remains competitive in the evaluated zero-delay settings and exhibits favorable zero-shot cross-difficulty generalization on Cooperative Navigation.

1 INTRODUCTION

Communication is central to coordination in cooperative multi-agent reinforcement learning (MARL) under partial observability. Most communication methods assume that messages are available within the current decision step. In practical systems, sensing, computation, and transmission latencies can induce cross-timestep communication delays, where a message generated at timestep t reaches the receiver only at $t+d$. As illustrated in Fig. 1, conditioning on a delayed message rather than its timely reference can shift the receiver's decision context and change the resulting transition. The key question is therefore whether a received delayed message still provides positive utility at consumption time.

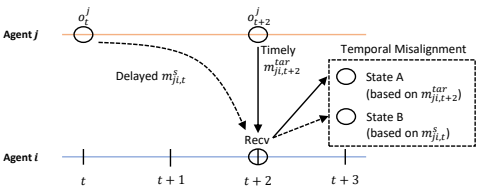


Figure 1: Temporal misalignment under cross-timestep communication delays.

Prior MARL communication methods mainly study delay-free settings, including centralized protocols and selective schemes. Delayed-communication methods such as DACOM (Yuan et al., 2023) and CoDe (Song et al., 2025) explicitly consider latency, but they do not directly score delayed-message utility through both no-message and timely-reference counterfactuals. This leaves open how a receiver should decide whether a delayed message remains useful under its current local context.

We address this gap with a gain–cost view of delayed-message utility. We first formalize cooperative MARL with cross-timestep communication delays as a delayed-communication partially observable Markov game (DeComm-POMG), where delayed delivery is part of the decision process. We then derive a delay-induced return-loss bound that relates the return gap between policies using timely references and those using delayed messages to an accumulated information gap. This motivates two complementary counterfactual terms: communication gain compares using a delayed message

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

against receiving no message, whereas delay cost compares it against the timely reference through an information-gap surrogate for delay-induced return loss. Together, they define Communication Gain and Delay Cost (CGDC), a gain–cost metric for evaluating delayed-message utility under centralized training with decentralized execution (CTDE).

Building on CGDC, we propose CDCMA, an actor–critic framework for communication under cross-timestep delays. It realizes the gain–cost view through three components. The Delay Communication Objective Selector (DCOS) learns critic-based CGDC targets and predicts per-pair scores for partner selection. The Observation Trajectory Generator (OTG) constructs future-aware outgoing messages to reduce consumption-time misalignment. The CGDC-guided Attentional Message Aggregator (CAMA) aggregates received delayed messages using CGDC-guided attention. We evaluate CDCMA on partially observable variants of Cooperative Navigation and Predator Prey without teammate vision, and on benchmark maps from the StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al., 2019), under bounded stochastic delay regimes. Results show that CDCMA achieves higher mean performance than the communication baselines across the bounded delayed settings studied here and retains favorable zero-shot performance across unseen delay difficulties on Cooperative Navigation.

Our contributions are summarized as follows:

- We formalize cooperative MARL with cross-timestep communication delays as **DeComm-POMG**, which models delayed message arrivals and admits the augmented-state Markov representation used in our analysis.
- We introduce **CGDC**, a gain–cost metric that evaluates delayed-message utility through no-message and timely-reference counterfactuals, and derive a delay-induced return-loss bound based on the information gap between action distributions under timely-reference and delayed-message inputs.
- We propose **CDCMA**, which realizes CGDC through partner selection, future-aware message construction, and CGDC-guided delayed-message aggregation, and validate it across bounded stochastic delay regimes.

2 Related Work

Most MARL communication methods are developed for delay-free settings and can be broadly grouped into centralized and selective communication. Centralized protocols broadcast or aggregate messages system-wide, often through differentiable message passing, as in CommNet (Sukhbaatar et al., 2016), DIAL (Foerster et al., 2016), and BiCNet (Peng et al., 2017); later work improves expressiveness or robustness through recurrent aggregation, targeted communication, message filtering, and auxiliary objectives (Wang & Sartoretti, 2022; Das et al., 2019; Kim et al., 2019b; Guan et al., 2022). Selective methods improve scalability by restricting communication to selected agents, entities, or timesteps, as in ATOC (Jiang & Lu, 2018), I2C (Ding et al., 2020), SMS (Xue et al., 2022a), G2ANet (Liu et al., 2020), T2MAC (Sun et al., 2024), TGCNet (Zhang et al., 2025), and GRDC (Gao et al., 2026). Although these methods improve communication efficiency, they generally assume that exchanged messages are available within the current decision step.

A smaller line of work studies communication with delayed message arrivals. DACOM (Yuan et al., 2023) is delay-aware and uses TimeNet to adjust how long an agent waits for messages within a decision step, while CoDe (Song et al., 2025) infers long-term intents via future-action prediction and fuses asynchronous messages through dual alignment. Complementary work studies observation delays: RDC (Fu et al., 2025) compensates delayed observation components under stochastic individual delays. Our work instead focuses on delayed-message utility under unobserved cross-timestep communication delays, evaluating received delayed messages through no-message and timely-reference counterfactuals. Further discussion appears in Appendix B.

3 Problem Formulation and Gain–Cost View

3.1 DeComm-POMG

We build on the standard partially observable Markov game (POMG) formulation (Littman, 1994) and extend it to cross-timestep communication delays. Unlike standard communication-augmented

POMGs, a message generated at timestep t may become available only after several timesteps, so delayed delivery must be treated as part of the decision process.

Definition 3.1 (DeComm-POMG)**.** A **Delayed-Communication Partially Observable Markov Game** is the tuple $(\mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{R}, \mathcal{P}, \mathcal{M}, \mathcal{D}, \rho, \gamma, \varphi)$, where the first seven components coincide with a standard communication-augmented POMG, $\mathcal{D} = \{\mathcal{D}_{ji} \subset \mathbb{Z}_{\geq 0} \mid j \neq i\}$ specifies per-pair delay domains, ρ is the initial-state distribution, $\gamma \in (0, 1)$ is the discount factor, and $\varphi = \{\varphi_{ji} \mid j \neq i\}$ governs the delay process. At each timestep t, a delay $d_{ji,t} \in \mathcal{D}_{ji}$ is sampled for each link $(j \to i)$ according to φ_{ji}. We assume that the delay process is exogenous and Markov, with any latent delay-process state included in s_t, and that its realizations are not observable to agents.

At each timestep, agents choose an environment action $\boldsymbol{a}_t$ and outgoing messages $\boldsymbol{m}_t^s$. For the Markov representation below, we regard their combination $\boldsymbol{u}_t := (\boldsymbol{a}_t, \boldsymbol{m}_t^s)$ as the joint environment–communication action. Let $m_{ji,t}^s$ denote the message sent from agent j to agent i at timestep t, and let $m_{ij,t}^r$ denote the message from j that is available to agent i at receiver timestep t. If the sampled delay is $d_{ji,t}$, then $m_{ji,t}^s$ arrives at timestep $t+d_{ji,t}$. If multiple messages from the same sender arrive simultaneously, only the most recent one is retained: $\kappa_{ji}(t) \in \arg\max_{k \leq t}\{\, k \mid t = k + d_{ji,k} \,\}$ and $m_{ij,t}^r = m_{ji,\kappa_{ji}(t)}^s$.

3.2 Delayed-Message Utility: Communication Gain and Delay Cost

A received delayed message should be evaluated under two counterfactual comparisons. Compared with receiving no message, it may still improve decision making. Compared with the timely reference message, however, it may incur delay-related return loss because of temporal misalignment. This motivates a gain–cost view of delayed-message utility.

Markov representation. To analyze delayed communication with value-function tools, we augment the environment state with the message buffer $\mathcal{B}_t$ and write $\widetilde{s}_t = (s_t, \mathcal{B}_t)$, where $\mathcal{B}_t \in \mathcal{B}$ stores the in-flight and delivered messages required by the delayed-delivery process.

Proposition 3.2 (Markov representation of DeComm-POMG)**.** *With augmented state $\widetilde{s}_t = (s_t, \mathcal{B}_t)$ and joint environment–communication action $\boldsymbol{u}_t = (\boldsymbol{a}_t, \boldsymbol{m}_t^s)$, DeComm-POMG admits a controlled Markov representation on $\widetilde{\mathcal{S}} = \mathcal{S} \times \mathcal{B}$; specifically, $\Pr(\widetilde{s}_{t+1} \mid \widetilde{s}_{0:t}, \boldsymbol{u}_{0:t}) = \Pr(\widetilde{s}_{t+1} \mid \widetilde{s}_t, \boldsymbol{u}_t)$. The corresponding augmented POMG preserves the local observation and message histories of the original DeComm-POMG.*

Proposition 3.2 shows that delayed communication can be analyzed through an augmented Markov state. The proof is given in Appendix C.1. For the delay-loss analysis below, we hold the message-generation protocol fixed and compare the induced environment-action policies.

Communication gain. For agent i, the discounted return in DeComm-POMG is $J_i(\boldsymbol{\pi}) := \mathbb{E}_{\boldsymbol{\pi}}[\sum_{t \geq 0} \gamma^t r_i(\widetilde{s}_t, \boldsymbol{a}_t)]$. Let $\boldsymbol{\pi}^{(+j)}$ denote the joint policy in which agent i can use messages from sender j while other incoming messages are masked, and let $\boldsymbol{\pi}^{(0)}$ denote the joint policy in which all incoming messages to agent i are masked. Then $J_i(\boldsymbol{\pi}^{(+j)}) - J_i(\boldsymbol{\pi}^{(0)})$ captures the return-level benefit attributable to sender j.

To express this comparison at a given state–action pair, define $Q_i^{\boldsymbol{\pi}}(\widetilde{s}_t, \boldsymbol{a}_t) := \mathbb{E}_{\boldsymbol{\pi}}[\sum_{k \geq 0} \gamma^k r_i(\widetilde{s}_{t+k}, \boldsymbol{a}_{t+k}) \mid \widetilde{s}_t, \boldsymbol{a}_t]$. We define the communication gain as

$$V_{ij,t}^g(\widetilde{s}_t, \boldsymbol{a}_t) := Q_i^{(+j)}(\widetilde{s}_t, \boldsymbol{a}_t) - Q_i^{(0)}(\widetilde{s}_t, \boldsymbol{a}_t). \tag{1}$$

This compares continuation values under the two message-access policies, conditional on the same current state and joint action. With bounded rewards and $\gamma \in (0, 1)$, $V_{ij,t}^g$ is uniformly bounded; the bound is stated in Appendix C.2.

Delay-induced return loss and information gap. Communication gain alone is insufficient, because a delayed message may be useful relative to no message yet inferior to its timely reference. Let $m_{ji,t}^{\mathrm{tar}}$ denote the timely reference for sender j at receiver timestep t, representing the sender-side information aligned with the message-consumption time t. Its concrete training-time instantiation is

given in Sec. 3.3. For this policy comparison, when no message from sender j is available at timestep t, both the received and timely-reference conditions are set to $m^{\emptyset}$.

Let $\boldsymbol{\pi}^{\text{delayed}}$ denote the induced environment-action policy in which agent i conditions on the received delayed message $m^r_{ij,t}$, and let $\boldsymbol{\pi}^{\text{timely}}$ be identical except that agent i conditions on the timely reference $m^{\text{tar}}_{ji,t}$. Both policies share the same message-generation protocol and differ only in this action-conditioning input. We define the potential delay-induced return loss as

$$\mathcal{L}_i^{\text{delay}} := \left[J_i(\boldsymbol{\pi}^{\text{timely}}) - J_i(\boldsymbol{\pi}^{\text{delayed}})\right]_+, \tag{2}$$

where $[x]_+ := \max\{x, 0\}$ accounts for the fact that the timely condition need not yield a larger return on every policy instance.

To relate this return-level loss to a local discrepancy, fix a receiver timestep t and hold fixed the information available before consuming incoming messages. Let $x_{i,t}$ denote this message-free local input of agent i. We define the information gap as

$$\Delta_{\text{info}}(i, j, t) = D_{\text{KL}}\big(\pi_i(\cdot \mid x_{i,t}, m^{\text{tar}}_{ji,t}) \,\big\|\, \pi_i(\cdot \mid x_{i,t}, m^r_{ij,t})\big). \tag{3}$$

This quantity measures the action-distribution gap induced by replacing the timely reference condition with the received delayed message under the same local input.

Theorem 3.3 (Delay-induced return-loss bound)**.** *Assume $|r_i(\widetilde{s}, \boldsymbol{a})| \le R_{\max}$, $\gamma \in (0, 1)$, and that the decentralized joint environment-action policy factorizes across agents. Then*

$$\mathcal{L}_i^{\text{delay}} \;\le\; \frac{2R_{\max}}{1-\gamma}\, \mathbb{E}_{\tau\sim\boldsymbol{\pi}^{\text{timely}}}\left[\sum_{t\ge 0}\gamma^t \sqrt{\frac{1}{2}\Delta_{\text{info}}(i, j, t)}\right], \tag{4}$$

where the expectation is taken over trajectories τ under $\boldsymbol{\pi}^{\text{timely}}$ on augmented DeComm-POMG.

The bound relates potential delay-induced return loss to the accumulated information gap between action distributions under timely-reference and delayed-message inputs. Together with Eq. (1), it motivates a gain–cost view of delayed-message utility: useful delayed communication should have high utility relative to no message and low loss relative to the timely reference.

3.3 Tractable Surrogates and CGDC

Under CTDE, we instantiate this gain–cost view with two critics: Q_i^{full} provides the actor–critic learning signal, while the auxiliary Q_i^{gain} constructs the critic-induced tempered policy and the gain surrogate used for CGDC targets; decentralized execution uses only local inputs.

Critic-induced tempered policy. The information gap in Eq. (3) is a theoretical quantity defined by comparing actor policies under timely-reference and delayed-message conditions. To construct a tractable training-time delay-cost surrogate, we introduce a critic-induced tempered policy as a discrepancy probe under a common value reference. This probe is used only during CTDE training; execution does not access the timely reference. Let $\boldsymbol{o}_{-j,t}$ and $\boldsymbol{a}_{-j,t}$ denote the observations and actions of all agents except agent j at timestep t, and let $\boldsymbol{a}_{-ij,t}$ denote the actions of all agents except agents i and j. Given a message condition m for agent i at t, we define

$$\widetilde{\pi}_i(m) = \text{softmax}_{a_{i,t}\in\mathcal{A}_i}\Big[\eta\, Q_i^{\text{gain}}\big(a_{i,t}, \boldsymbol{o}_{-j,t}, \boldsymbol{a}_{-ij,t}, m\big)\Big], \tag{5}$$

where $\eta > 0$ controls the sharpness of the softmax.

Delay cost surrogate. For pairwise CGDC targets, centralized training constructs a delayed-message condition for every $j \neq i$, including pairs without actual arrivals (Appendix D.3). Within these target evaluations, $m^r_{ij,t}$ denotes the constructed reception condition, with timely reference $m^{\text{tar}}_{ji,t} := e_{j,t}$, the sender's current trajectory embedding computed without future prediction. While OTG compensates for temporal evolution, $e_{j,t}$ provides the consumption-time reference; neither the constructed conditions nor the reference are available to the execution policy. We define the delay cost surrogate as

$$V^c_{ij,t} = D_{\text{KL}}\Big(\widetilde{\pi}_i(m^{\text{tar}}_{ji,t}) \,\Big\|\, \widetilde{\pi}_i(m^r_{ij,t})\Big) \;\ge 0. \tag{6}$$

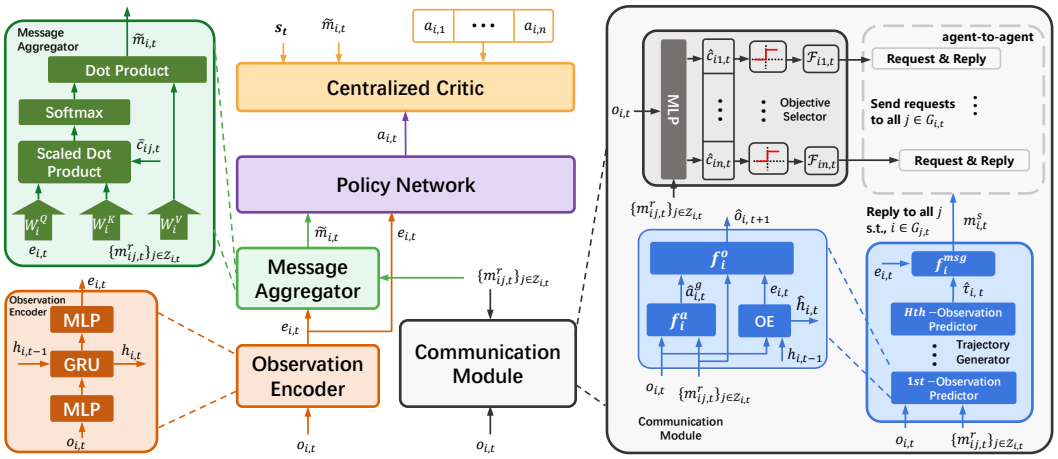


Figure 2: Overall architecture of CDCMA.

For the following corollary, evaluate this discrepancy under the message conditions of Theorem 3.3, including its empty-message convention. Write $\pi_i^{\mathrm{tar}} := \pi_i(\cdot \mid x_{i,t}, m_{ji,t}^{\mathrm{tar}})$, $\pi_i^{\mathrm{del}} := \pi_i(\cdot \mid x_{i,t}, m_{ij,t}^{r})$, $\widetilde{\pi}_i^{\mathrm{tar}} := \widetilde{\pi}_i(m_{ji,t}^{\mathrm{tar}})$, and $\widetilde{\pi}_i^{\mathrm{del}} := \widetilde{\pi}_i(m_{ij,t}^{r})$. Define the actor–probe discrepancies

$$\delta_{ij,t}^{\mathrm{tar}} := \|\pi_i^{\mathrm{tar}} - \widetilde{\pi}_i^{\mathrm{tar}}\|_{\mathrm{TV}}, \qquad \delta_{ij,t}^{\mathrm{del}} := \|\pi_i^{\mathrm{del}} - \widetilde{\pi}_i^{\mathrm{del}}\|_{\mathrm{TV}}. \tag{7}$$

Corollary 3.4 (Return-loss bound with critic-tempered delay cost)**.** *Under the conditions of Theorem 3.3,*

$$\mathcal{L}_i^{\mathrm{delay}} \le \frac{2R_{\max}}{1-\gamma}\mathbb{E}_{\tau\sim\boldsymbol{\pi}^{\mathrm{timely}}}\left[\sum_{t\ge 0}\gamma^t\left(\delta_{ij,t}^{\mathrm{tar}} + \sqrt{\frac{1}{2}V_{ij,t}^{\mathrm{c}}} + \delta_{ij,t}^{\mathrm{del}}\right)\right]. \tag{8}$$

The corollary separates the implemented critic-tempered discrepancy from its actor–probe approximation terms. The proof and further discussion are given in Appendix D.1.

Gain surrogate. For the pairwise gain target, we evaluate the auxiliary gain critic with and without the message from sender j. Let $m^{\emptyset}$ denote a null message representing message absence. We define the gain surrogate as

$$\widehat{V}_{ij,t}^{g} = Q_i^{\mathrm{gain}}\big(a_{i,t}, \boldsymbol{o}_{-j,t}, \boldsymbol{a}_{-ij,t}, m_{ij,t}^{r}\big) - Q_i^{\mathrm{gain}}\big(a_{i,t}, \boldsymbol{o}_{-j,t}, \boldsymbol{a}_{-ij,t}, m^{\emptyset}\big), \tag{9}$$

Motivated by Eq. (1), this surrogate measures the message-conditioned difference in learned value predictions. The two critic branches share continuation actions during training; their targets are specified in Appendix D.3.

CGDC. We combine communication gain and delay cost into a single gain–cost score.

Definition 3.5 (CGDC)**.** Let $V_{ij,t}^{g}$ and $V_{ij,t}^{c}$ be defined in Eqs. (1) and (6). The **Communication Gain and Delay Cost** is

$$c_{ij,t} = V_{ij,t}^{g} - \lambda\, V_{ij,t}^{c}, \tag{10}$$

where $\lambda \ge 0$ balances the two terms and controls the conservativeness of the communication gate.

For training, we instantiate Eq. (10) with $\widehat{V}_{ij,t}^{g}$ in place of $V_{ij,t}^{g}$. Both terms evaluate the same pairwise delayed-message condition, relative to message absence and the timely reference, respectively.

4 CDCMA: A Realization of the Gain–Cost View

4.1 Overview

The gain–cost view in Sec. 3 suggests three requirements for communication under cross-timestep delays: messages should be requested when their predicted CGDC scores are positive, outgoing

messages should remain informative at future consumption time, and received delayed messages should be aggregated using content relevance and the request-time predicted CGDC scores stored with them. CDCMA realizes these requirements through a *request–construct–aggregate* pipeline.

At timestep t, agent i first encodes its local observation into a trajectory embedding and predicts which teammates to query. Each sender constructs one outgoing message by combining historical trajectory information with predicted future observations, and the same message is used for all receivers that request it. Finally, because the messages available at receiver timestep t generally differ from those queried at t, agent i aggregates received delayed messages and selects its action from the aggregated message and local embedding.

This pipeline is illustrated in Fig. 2. CDCMA consists of an observation encoder, the *Delay Communication Objective Selector* (DCOS), the *Observation Trajectory Generator* (OTG), the *CGDC-guided Attentional Message Aggregator* (CAMA), a decentralized policy network, and two critics used during training. DCOS, OTG, and CAMA implement partner selection, future-aware message construction, and CGDC-guided delayed-message aggregation, respectively. Pseudocode and complexity are deferred to Algorithm 1 and Appendix D.4. Training details are provided in Appendix D.3.

4.2 Delay Communication Objective Selector

Under cross-timestep communication delays, a receiver must decide whom to query before future arrivals are known. DCOS addresses this problem by predicting a per-pair CGDC score from local context available at execution and requesting a message only when the predicted score is positive.

Let $\mathcal{Z}_{i,t} \triangleq \{\, j \neq i \mid m^r_{ij,t} \neq m^{\emptyset} \,\}$ denote the available-sender set of agent i at timestep t, where $m^{\emptyset}$ is a null message. Given the current local observation and the received-message context $\{m^r_{ij,t}\}_{j\in\mathcal{Z}_{i,t}}$, agent i predicts the scores $\{\hat{c}_{ij,t}\}_{j\neq i}$ by

$$\{\hat{c}_{ij,t}\}_{j\neq i} = \phi_i\big(o_{i,t}, \{m^r_{ij,t}\}_{j\in\mathcal{Z}_{i,t}}; \theta_{\phi_i}\big), \qquad \hat{c}_{ii,t} = 0. \tag{11}$$

Here, $\hat{c}_{ij,t}$ is a learned estimator of the CGDC score defined in Sec. 3.3. Thresholding the prediction yields the request mask $F_{ij,t} = \mathbb{I}[\hat{c}_{ij,t} > 0]$ and the queried partner set $\mathcal{G}_{i,t} = \{\, j \neq i \mid F_{ij,t} = 1 \,\}$.

The distinction between $\mathcal{G}_{i,t}$ and $\mathcal{Z}_{i,t}$ is essential. $\mathcal{G}_{i,t}$ is the set of teammates queried at timestep t, whereas $\mathcal{Z}_{i,t}$ is the set of senders whose delayed messages are available at timestep t. Because delivery is delayed and unobserved, these two sets generally differ. This mismatch motivates the remaining components: OTG constructs future-aware outgoing messages, while CAMA aggregates received delayed messages.

4.3 Observation Trajectory Generator

A message generated at timestep t may be consumed several timesteps later, so a sender representation based only on the current observation may be misaligned when consumed. OTG mitigates this temporal misalignment by constructing future-aware outgoing messages from predicted observations.

Let H denote the prediction horizon. At t, OTG outputs a predicted observation trajectory $\hat{\tau}_{i,t} = (\hat{o}_{i,t+1}, \ldots, \hat{o}_{i,t+H})$. The outgoing message is then formed by encoding the combined trajectory $(\tau_{i,t}, \hat{\tau}_{i,t})$, where $\tau_{i,t}$ denotes the historical trajectory summary of agent i up to t.

Because future observations depend on both teammates' actions and local dynamics, OTG factorizes prediction into two parts. First, a teammate-action predictor estimates the actions of the agents whose messages are currently available:

$$\hat{a}^g_{i,t} := f^a_i\big(o_{i,t}, \{m^r_{ij,t}\}_{j\in\mathcal{Z}_{i,t}}\big) = \{\hat{a}_{j,t}\}_{j\in\mathcal{Z}_{i,t}}. \tag{12}$$

Second, an observation-dynamics predictor estimates the next-step observation residual:

$$\hat{o}_{i,t+1} = o_{i,t} + f^o_i\big(e_{i,t}, \{m^r_{ij,t}\}_{j\in\mathcal{Z}_{i,t}}, \hat{a}^g_{i,t}\big), \tag{13}$$

where $e_{i,t}$ denotes the local trajectory embedding. Recursively applying Eqs. (12)–(13) yields the H-step predicted trajectory $\hat{\tau}_{i,t}$.

OTG therefore constructs a future-aware outgoing message rather than simply encoding the sender's current state, keeping transmitted information better aligned with the receiver's decision context at consumption time.

4.4 CGDC-GUIDED ATTENTIONAL MESSAGE AGGREGATOR

At receiver timestep t, agent i may have multiple delayed messages available for decision making. CAMA combines content-based attention with the request-time CGDC scores stored with the received messages in $\mathcal{Z}_{i,t}$.

For each $j \in \mathcal{Z}_{i,t}$, let $\bar{c}_{ij,t} > 0$ denote the request-time predicted CGDC score stored with message $m^r_{ij,t}$. Given the local trajectory embedding $e_{i,t}$, CAMA computes $q_{i,t} = W_i^Q e_{i,t}$, $k_{ij,t} = W_i^K m^r_{ij,t}$, and $v_{ij,t} = W_i^V m^r_{ij,t}$ for $j \in \mathcal{Z}_{i,t}$, where W_i^Q, W_i^K, and W_i^V are learnable projections. It then modulates standard content-based attention by the stored CGDC score:

$$\alpha^{\mathrm{cgdc}}_{ij,t} = \frac{\bar{c}_{ij,t} \exp\big(\beta\, q_{i,t}^\top k_{ij,t}\big)}{\sum_{l \in \mathcal{Z}_{i,t}} \bar{c}_{il,t} \exp\big(\beta\, q_{i,t}^\top k_{il,t}\big)}, \qquad j \in \mathcal{Z}_{i,t}, \tag{14}$$

where $\beta > 0$ controls attention sharpness. The stored CGDC score serves as a utility prior rather than a fixed aggregation weight, while the final attention also depends on the current receiver query and message content. Appendix D.2 shows that the weights in Eq. (14) uniquely maximize a KL-regularized attention objective.

The aggregated delayed-message representation is $\tilde{m}_{i,t} = \sum_{j \in \mathcal{Z}_{i,t}} \alpha^{\mathrm{cgdc}}_{ij,t} v_{ij,t}$; if $\mathcal{Z}_{i,t} = \varnothing$, we define $\tilde{m}_{i,t} = \mathbf{0}$. Thus, CAMA preserves content discrimination while using CGDC as a utility prior.

5 EXPERIMENTS

We evaluate CDCMA under four bounded stochastic delay regimes and a delay-free setting. The experiments examine three aspects: performance against communication baselines, the contribution of each component and key hyperparameter, and behavior under delay-free and zero-shot cross-difficulty settings. Implementation details and additional results, including MMM2, appear in Appendix E.

5.1 EXPERIMENTAL SETUP

Benchmarks and metrics. We evaluate CDCMA on variants of Cooperative Navigation (CN) and Predator Prey (PP) without teammate vision in Multi-Agent Particle Environment (MPE), and on two communication-sensitive SMAC maps: `1o_2r_vs_4r` and `1o_10b_vs_1r`. We report mean episode reward on MPE and win rate on SMAC. Performance tables report mean and standard deviation across four independent training seeds; learning-curve shading denotes one standard deviation. Evaluation and curve-smoothing details are given in Appendix E.1.

Baselines. We compare CDCMA with CoDe (Song et al., 2025), DACOM (Yuan et al., 2023), TGCNet (Zhang et al., 2025), T2MAC (Sun et al., 2024), SMS (Xue et al., 2022a), G2ANet (Liu et al., 2020), and ATOC (Jiang & Lu, 2018). All methods share the same delay distributions and message-delivery rules. The baselines use official code and hyperparameters, except CoDe, which we reimplement. CoDe uses framework-provided message ages; CDCMA does not. Throughout the experiments, DACOM denotes the variant without TimeNet. Implementation and training details appear in Appendix E.1.

Delay settings. For each ordered pair (j, i) and send timestep k, the communication delay is an integer $d_{ji,k} \in \{1, \ldots, d_{\max}\}$ with $d_{\max} = 5$, drawn from a difficulty-indexed probability mass function $p_\ell(d)$, where $\ell \in \{\texttt{easy}, \texttt{medium}, \texttt{hard}, \texttt{super_hard}\}$. We also report a delay-free setting with $d_{ji,k} = 0$. The exact delay PMFs are given in Appendix E.2, and the evaluation protocol, implementation details, and hyperparameters are given in Appendix E.1.

5.2 MAIN RESULTS

Table 1 reports performance across tasks and delay settings. CDCMA achieves the highest mean performance on all four tasks under all four bounded stochastic delay regimes, outperforming CoDe and the TimeNet-free DACOM variant. Its advantage persists across delay difficulties, although the margin varies by task and baseline. For example, its advantage over CoDe on CN grows from 1.41 under `easy` to 2.53 under `super_hard`, while that on `1o_10b_vs_1r` remains above 48 percentage points throughout. CDCMA also remains ahead of strong baselines originally designed for delay-free

Table 1: Mean episode reward (win rate) across tasks under four bounded stochastic delay regimes. The highest mean is in bold.

Task	Difficulty	CDCMA	CoDe	DACOM	TGCNet	T2MAC	SMS	G2ANet	ATOC
Cooperative Navigation	easy	**-1.70**(0.07)	-3.11(0.14)	-3.16(0.43)	-2.08(0.32)	-2.23(0.04)	-2.57(0.11)	-2.24(0.09)	-2.51(0.62)
	medium	**-1.80**(0.36)	-3.09(0.05)	-3.14(0.13)	-2.21(0.35)	-2.27(0.08)	-2.67(1.01)	-2.48(0.07)	-2.43(0.14)
	hard	**-1.78**(0.22)	-4.36(2.55)	-3.52(1.02)	-2.35(0.34)	-2.51(0.44)	-2.81(0.47)	-2.71(0.07)	-2.75(0.22)
	super_hard	**-1.89**(0.13)	-4.42(2.75)	-3.00(0.21)	-2.56(0.36)	-2.53(0.17)	-3.04(0.64)	-2.94(0.06)	-2.79(0.45)
Predator Prey	easy	**-0.93**(0.03)	-1.82(0.03)	-1.97(0.23)	-1.53(0.14)	-1.10(0.15)	-1.46(0.27)	-1.17(0.01)	-2.38(0.38)
	medium	**-0.91**(0.04)	-1.83(0.03)	-2.19(0.42)	-1.60(0.14)	-1.27(0.22)	-1.36(0.17)	-1.40(0.05)	-2.75(0.54)
	hard	**-1.02**(0.06)	-1.86(0.08)	-1.93(0.01)	-1.72(0.15)	-1.47(0.39)	-1.51(0.06)	-1.71(0.08)	-2.98(0.19)
	super_hard	**-0.97**(0.10)	-1.85(0.07)	-2.01(0.04)	-1.75(0.11)	-1.66(0.33)	-1.78(0.71)	-1.85(0.04)	-3.08(0.86)
1o_2r_vs_4r(%)	easy	**87.50**(9.88)	58.59(14.29)	39.84(3.93)	63.28(14.29)	37.50(27.83)	48.44(17.02)	45.31(12.88)	14.06(9.72)
	medium	**75.00**(8.84)	40.63(7.65)	35.94(6.51)	56.25(7.65)	38.28(28.91)	50.00(10.83)	21.88(5.10)	24.22(17.75)
	hard	**68.75**(13.74)	28.91(9.33)	39.84(3.93)	50.78(9.33)	33.59(31.18)	38.28(2.99)	28.13(16.73)	11.72(17.75)
	super_hard	**63.28**(11.23)	24.22(6.93)	38.28(6.93)	49.22(6.93)	20.31(22.46)	40.63(14.88)	33.59(10.94)	17.19(12.10)
1o_10b_vs_1r(%)	easy	**92.19**(4.03)	32.81(1.80)	11.72(4.69)	66.41(6.44)	33.59(38.98)	13.28(17.75)	44.53(25.94)	0.00(0.00)
	medium	**86.72**(6.93)	35.16(4.69)	10.16(8.22)	42.97(2.99)	28.91(34.55)	10.16(20.31)	23.44(10.97)	0.00(0.00)
	hard	**83.59**(6.44)	24.22(22.88)	10.94(6.51)	54.69(14.09)	31.25(28.30)	0.00(0.00)	8.59(8.98)	0.00(0.00)
	super_hard	**73.44**(5.41)	25.00(15.31)	22.66(8.61)	44.53(10.33)	25.00(29.87)	0.00(0.00)	27.34(6.93)	0.00(0.00)

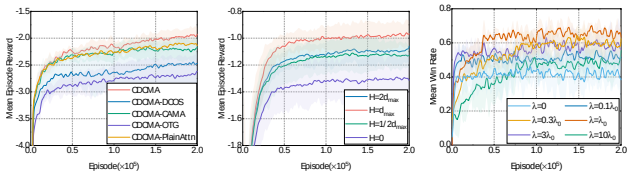
communication, particularly TGCNet on SMAC and CN and T2MAC on PP. Against TGCNet under super_hard, CDCMA achieves 73.44% versus 44.53% on 1o_10b_vs_1r and 63.28% versus 49.22% on 1o_2r_vs_4r. MPE and SMAC learning curves appear in Appendix E.3.

These patterns are consistent with CDCMA's design: DCOS filters communication using predicted CGDC, OTG reduces consumption-time misalignment through future-aware messages, and CAMA uses CGDC as a utility prior for delayed-message aggregation.

5.3 Ablations and Sensitivity

We evaluate module ablations on CN, horizon sensitivity on PP, and gain–cost weight sensitivity on 1o_2r_vs_4r. Unless otherwise noted, all ablations are conducted under the super_hard delay regime.

Module ablations. We ablate the three main components of CDCMA on CN: *CDCMA-OTG* (without OTG), *CDCMA-CAMA* (without CAMA), *CDCMA-DCOS* (without DCOS), and *CDCMA-PlainAttn* (replacing CAMA with standard dot-product attention). As shown in Fig. 3a, removing any component degrades performance. The largest drop in this setting occurs without OTG, indicating that future-aware outgoing messages provide the largest contribution among the evaluated module ablations. *CDCMA-PlainAttn* also underperforms CDCMA, showing that CAMA benefits from combining content attention with CGDC-based utility priors.



(a) Module ablations (b) Horizon H (c) CGDC weight λ

Figure 3: Ablations and sensitivity analyses.

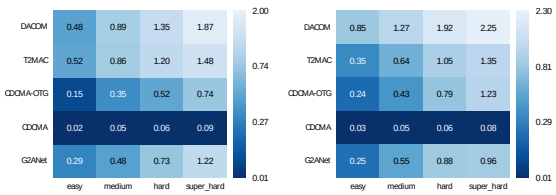
Sensitivity to the prediction horizon H. We vary $H \in \{0, \lceil \frac{1}{2} d_{\max} \rceil, d_{\max}, 2d_{\max}\}$ on PP under super_hard delays. Compared with $H = 0$ (no future prediction), performance improves up to $H = d_{\max}$ but deteriorates beyond this value. This pattern is consistent with a coverage–rollout-error trade-off: increasing H helps outgoing messages cover typical delays, whereas longer rollouts accumulate larger multi-step prediction errors. Qualitative OTG diagnostics are provided in Appendix E.6.

Sensitivity to the CGDC weight λ. On 1o_2r_vs_4r under super_hard, we sweep $\lambda \in \{0, 0.1\lambda_0, 0.3\lambda_0, \lambda_0, 3\lambda_0, 10\lambda_0\}$, where λ_0 matches the gain and cost scales within each training minibatch; its definition and numerical safeguards are given in Appendix D.3. Unless otherwise stated, we use $\lambda = \lambda_0$. Here, $\lambda = 0$ is the gain-only variant without the delay-cost penalty. Fig. 3c shows that both this variant and overly large λ are suboptimal, whereas a moderate range around λ_0 performs best. This pattern is consistent with the intended gain–cost balance: small λ underweights delay cost, whereas large λ can suppress otherwise useful communication.

5.4 Information-Gap Analysis and Robustness

We finally examine whether CDCMA reduces the decision-distribution gap and whether its performance persists under delay-free and zero-shot cross-difficulty settings.

Decision-distribution information gaps. Motivated by Sec. 3.2, we compare decision distributions under timely-reference and delayed-message conditions, using actor outputs or softmax-normalized action values as appropriate for each method. The timely branch replaces all received messages with the corresponding senders' current embeddings, keeping the network and all other inputs fixed. Actions use delayed messages; the timely branch is used only for diagnosis. Appendix E.7 defines the metric and averaging.



(a) Predator Prey (b) `1o_10b_vs_1r`

Figure 4: Decision-distribution information gaps.

We report gaps averaged over all agents and timesteps on PP and `1o_10b_vs_1r` under `easy`–`super_hard`. Figure 4 shows smaller decision-distribution gaps for CDCMA than the compared methods across all regimes, while its gap increases moderately with delay difficulty. Removing OTG yields larger gaps, consistent with its role in mitigating temporal misalignment. Together with Table 1 and Fig. 3a, these results associate smaller decision-distribution gaps with stronger performance under the evaluated delays.

Delay-free evaluation. To examine behavior in the zero-delay limit, we evaluate `1o_10b_vs_1r` with $d_{ji,k} = 0$ and set $H = 0$ accordingly. CDCMA remains stable and competitive across seeds in this zero-delay setting. The corresponding learning curve is provided in Appendix E.4.

Zero-shot cross-difficulty. To test robustness under delay-distribution shift, we train on CN under `easy` and test zero-shot under `medium`, `hard`, and `super_hard`. Table 2 shows that CDCMA achieves the best mean episode reward under all three unseen delay difficulties and exhibits relatively modest degradation as the delay distribution becomes harder. From `easy` to `super_hard`, its mean reward decreases by 0.23, compared with 0.91 for CoDe, 0.57 for TGCNet, and 0.51 for T2MAC.

Table 2: CN zero-shot cross-difficulty generalization. Train: `easy`; entries are mean(std) episode reward.

Method	Train	Test		
	`easy`	`medium`	`hard`	`super_hard`
CDCMA	**-1.70(0.07)**	**-1.82(0.15)**	**-1.84(0.19)**	**-1.93(0.09)**
CoDe	−3.11(0.14)	−3.42(0.19)	−3.59(0.71)	−4.02(1.88)
DACOM	−3.16(0.43)	−3.21(0.27)	−3.37(0.39)	−3.29(0.45)
TGCNet	−2.08(0.32)	−2.28(0.33)	−2.41(0.41)	−2.65(0.47)
T2MAC	−2.23(0.04)	−2.31(0.12)	−2.62(0.55)	−2.74(0.59)
SMS	−2.57(0.11)	−2.71(0.94)	−2.89(0.97)	−3.11(0.77)
G2ANet	−2.24(0.09)	−2.57(0.11)	−2.85(0.09)	−3.01(0.13)
ATOC	−2.51(0.62)	−2.55(0.26)	−2.87(0.19)	−2.93(0.37)

6 Conclusion

We studied cooperative MARL under cross-timestep communication delays, where messages may be informative yet temporally misaligned when consumed. We formalized it as DeComm-POMG, introducing CGDC to assess delayed-message utility using no-message and timely-reference counterfactuals, deriving a delay-induced return-loss bound in terms of accumulated information gap.

We instantiated this view in CDCMA through CGDC-based partner selection, future-aware message construction, and CGDC-guided delayed-message aggregation. On MPE tasks without teammate vision and on SMAC maps, CDCMA achieves higher mean performance than communication baselines under the evaluated bounded stochastic delays. On the two representative tasks it yields smaller decision-distribution gaps. CDCMA remains competitive in the evaluated zero-delay settings and retains favorable zero-shot performance across unseen delay difficulties on Cooperative Navigation.

Limitations and future work Our study assumes bounded, discrete, reliably delivered, exogenous message delays, instantaneous requests, and a fixed OTG prediction horizon. Extending the framework to nonstationary latency, packet loss, bandwidth constraints, and larger, noisier, or more stochastic environments remains important future work.

AI USE STATEMENT

Generative AI tools were used to improve the grammar and readability of the manuscript. All AI-assisted edits were reviewed by the authors, who take full responsibility for the final content.

REFERENCES

Akshat Agarwal, Sumit Kumar, Katia Sycara, and Michael Lewis. Learning transferable cooperative behavior in multi-agent teams. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1741–1743, 2020.

Tianshu Chu, Sandeep Chinchali, and Sachin Katti. Multi-agent reinforcement learning for networked system control. In *International Conference on Learning Representations*, 2020.

Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. In *International Conference on Machine Learning*, pp. 1538–1546, 2019.

Ziluo Ding, Tiejun Huang, and Zongqing Lu. Learning individually inferred communication for multi-agent cooperation. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22069–22079, 2020.

Ziluo Ding, Zeyuan Liu, Zhirui Fang, Kefan Su, Liwen Zhu, and Zongqing Lu. Multi-agent coordination via multi-level communication. In *Advances in Neural Information Processing Systems*, volume 37, pp. 118513–118539, 2024.

Yali Du, Bo Liu, Vincent Moens, Ziqi Liu, Zhicheng Ren, Jun Wang, Xu Chen, and Haifeng Zhang. Learning correlated communication topology in multi-agent reinforcement learning. In *Proc. 20th Int. Conf. Auton. Agents MultiAgent Syst.*, pp. 456–464, 2021.

Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 29, pp. 2145–2153, 2016.

Benjamin Freed, Guillaume Sartoretti, Jiaheng Hu, and Howie Choset. Communication learning via backpropagation in discrete channels with unknown noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7160–7168, 2020.

Songchen Fu, Siang Chen, Shaojing Zhao, Letian Bai, Hong Liang, Ta Li, and Yonghong Yan. Rainbow delay compensation: A multi-agent reinforcement learning framework for mitigating observation delays. *Advances in Neural Information Processing Systems*, 38:62039–62068, 2025.

Zihong Gao, Changsheng Qu, Yuanhui Hao, and Liangjun Ke. Communication in multiagent reinforcement learning via counterfactual message value. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 55(11):8152–8165, 2025. doi: 10.1109/TSMC.2025.3604230.

Zihong Gao, Hongjian Liang, Yuanhui Hao, Lei Hao, and Liangjun Ke. Grdc: A unified graph-driven framework for role discovery and communication in multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29441–29449, 2026.

Cong Guan, Feng Chen, Lei Yuan, Chenghe Wang, Hao Yin, Zongzhang Zhang, and Yang Yu. Efficient multi-agent communication via self-supervised information aggregation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1020–1033, 2022.

Guangzheng Hu, Yuanheng Zhu, Dongbin Zhao, Mengchen Zhao, and Jianye Hao. Event-triggered communication network with limited-bandwidth constraint for multi-agent reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):3966–3978, 2023. doi: 10.1109/TNNLS.2021.3135320.

Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation. In *Advances in Neural Information Processing Systems*, volume 31, pp. 7265–7275, 2018.

Daewoo Kim, Sangwoo Moon, David Hostallero, Wan Ju Kang, Taeyoung Lee, Kyunghwan Son, and Yung Yi. Learning to schedule communication in multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2019a.

Woojun Kim, Myungsik Cho, and Youngchul Sung. Message-dropout: An efficient training method for multi-agent deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 6079–6086, 2019b.

Woojun Kim, Jongeui Park, and Youngchul Sung. Communication in multi-agent reinforcement learning: Intention sharing. In *International Conference on Learning Representations*, 2020.

Toru Lin, Jacob Huh, Christopher Stauffer, Ser Nam Lim, and Phillip Isola. Learning to ground multi-agent communication with autoencoders. In *Advances in Neural Information Processing Systems*, volume 34, pp. 15230–15242, 2021.

Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning*, pp. 157–163, 1994.

Yong Liu, Weixun Wang, Yujing Hu, Jianye Hao, Xingguo Chen, and Yang Gao. Multi-agent game abstraction via graph attention neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7211–7218, 2020.

Zeyang Liu, Lipeng Wan, Xue Sui, Zhuoran Chen, Kewu Sun, and Xuguang Lan. Deep hierarchical communication graph in multi-agent reinforcement learning. In *IJCAI*, pp. 208–216, 2023.

Hangyu Mao, Zhengchao Zhang, Zhen Xiao, Zhibo Gong, and Yan Ni. Learning agent communication under limited bandwidth by message pruning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5142–5149, 2020.

Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 1495–1502, 2018.

Siddharth Nayak, Kenneth Choi, Wenqi Ding, Sydney Dolan, Karthik Gopalakrishnan, and Hamsa Balakrishnan. Scalable multi-agent reinforcement learning through intelligent information aggregation. In *International Conference on Machine Learning*, pp. 25817–25833, 2023.

Yaru Niu, Rohan Paleja, and Matthew Gombolay. Multi-agent graph-attention communication and teaming. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, pp. 964–973, 2021.

Peng Peng, Ying Wen, Yaodong Yang, Quan Yuan, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play StarCraft combat games, 2017. URL `https://arxiv.org/abs/1703.10069`. arXiv:1703.10069.

Zhiqiang Pu, Huimu Wang, Zhen Liu, Jianqiang Yi, and Shiguang Wu. Attention enhanced reinforcement learning for multi agent cooperation. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8235–8249, 2022. doi: 10.1109/TNNLS.2021.3116128.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft multi-agent challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2186–2188, 2019.

Shoucheng Song, Youfang Lin, Sheng Han, Chang Yao, Hao Wu, Shuo Wang, and Kai Lv. Code: Communication delay-tolerant multi-agent collaboration via dual alignment of intent and timeliness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23304–23312, 2025.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, volume 29, pp. 2244–2252, 2016.

Chuxiong Sun, Zehua Zang, Jiabao Li, Jiangmeng Li, Xiao Xu, Rui Wang, and Changwen Zheng. T2mac: Targeted and trusted multi-agent communication through selective engagement and evidence-driven integration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 15154–15163, 2024.

Yanchao Sun, Ruijie Zheng, Parisa Hassanzadeh, Yongyuan Liang, Soheil Feizi, Sumitra Ganesh, and Furong Huang. Certifiably robust policy learning against adversarial multi-agent communication. In *International Conference on Learning Representations*, 2023.

Rundong Wang, Xu He, Runsheng Yu, Wei Qiu, Bo An, and Zinovi Rabinovich. Learning efficient multi-agent communication: An information bottleneck approach. In *International Conference on Machine Learning*, pp. 9908–9918, 2020.

Tonghan Wang, Jianhao Wang, Chongyi Zheng, and Chongjie Zhang. Learning nearly decomposable value functions via communication minimization. In *International Conference on Learning Representations*, 2019.

Yutong Wang and Guillaume Sartoretti. Fcmnet: Full communication memory net for team-level cooperation in multi-agent systems. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 1355–1363, 2022.

Di Xue, Lei Yuan, Zongzhang Zhang, and Yang Yu. Efficient multi-agent communication via shapley message value. In *IJCAI*, pp. 578–584, 2022a.

Wanqi Xue, Wei Qiu, Bo An, Zinovi Rabinovich, Svetlana Obraztsova, and Chai Kiat Yeo. Mis-spoke or mis-lead: Achieving robustness in multi-agent communicative reinforcement learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 1418–1426, 2022b.

Lei Yuan, Jianhao Wang, Fuxiang Zhang, Chenghe Wang, Zongzhang Zhang, Yang Yu, and Chongjie Zhang. Multi-agent incentive communication via decentralized teammate modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 9466–9474, 2022.

Tingting Yuan, Hwei-Ming Chung, Jie Yuan, and Xiaoming Fu. Dacom: Learning delay-aware communication for multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11763–11771, 2023.

Sai Qian Zhang, Qi Zhang, and Jieyu Lin. Efficient communication in multi-agent reinforcement learning via variance based control. In *Advances in Neural Information Processing Systems*, volume 32, pp. 3235–3244, 2019.

Sai Qian Zhang, Qi Zhang, and Jieyu Lin. Succinct and robust multi-agent communication with temporal message control. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17271–17282, 2020.

Zhuohui Zhang, Bin He, Bin Cheng, and Gang Li. Bridging training and execution via dynamic directed graph-based communication in cooperative multi-agent systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23395–23403, 2025. doi: 10.1609/aaai.v39i22.34507. URL https://ojs.aaai.org/index.php/AAAI/article/view/34507.

A NOTATION SUMMARY

For convenience, Table 3 summarizes the main paper-specific symbols used in the formulation and method.

Table 3: Summary of the main paper-specific symbols.

Category	Symbol	Meaning
Delayed communication semantics	$\boldsymbol{u}_t$	Joint environment–communication action $(\boldsymbol{a}_t, \boldsymbol{m}_t^s)$
	$m^s_{ji,t}$	Message sent from agent j to agent i at timestep t
	$m_{ij,t}$	Message from agent j that is available to agent i at receiver timestep t
	$m^{\mathrm{tar}}_{ji,t}$	Timely reference condition for sender j at receiver timestep t
	$m^{\emptyset}$	Null / masked message input
	$d_{ji,t}$	Sampled communication delay for link $(j \to i)$ at timestep t
	$\mathcal{B}_t$	Message buffer at timestep t
	$\widetilde{s}_t$	Augmented state $(s_t, \mathcal{B}_t)$
	$x_{i,t}$	Message-free local input of agent i at timestep t
Gain, delay cost, and CGDC	$V^g_{ij,t}$	Communication gain from sender j to receiver i at timestep t
	$\widehat{V}^g_{ij,t}$	Critic-based surrogate of communication gain
	$\Delta_{\mathrm{info}}(i,j,t)$	Sender-specific actor-policy information gap
	$\Delta_{\mathrm{dec}}(i,t)$	Decision-distribution gap for the complete received-message collection
	$\mathcal{L}^{\mathrm{delay}}_i$	Potential delay-induced return loss
	$V^c_{ij,t}$	Critic-tempered delay-cost surrogate
	$\delta^{\mathrm{tar}}_{ij,t}$	Actor–probe discrepancy under the timely reference
	$\delta^{\mathrm{del}}_{ij,t}$	Actor–probe discrepancy under the delayed message
	$c_{ij,t}$	Communication Gain and Delay Cost (CGDC) score
	λ	Weight balancing communication gain and delay cost
CDCMA modules	ϕ_i	DCOS predictor for agent i
	$\hat{c}_{ij,t}$	Predicted CGDC score for communication from j to i
	$F_{ij,t}$	Binary request mask indicating whether agent i queries agent j
	$\mathcal{G}_{i,t}$	Set of senders queried by agent i at timestep t
	$\mathcal{Z}_{i,t}$	Set of senders whose messages are available to agent i at timestep t
	$e_{i,t}$	Local trajectory embedding of agent i at timestep t
	$\tau_{i,t}$	Historical trajectory summary of agent i up to timestep t
	$\hat{\tau}_{i,t}$	Predicted future observation trajectory of agent i
	H	OTG prediction horizon
	$\tilde{m}_{i,t}$	Aggregated delayed-message representation used by agent i
Critics and attention	Q^{full}_i	Centralized full critic used for actor–critic optimization
	Q^{gain}_i	Auxiliary gain critic used to construct gain/cost surrogates
	$\widetilde{\pi}_i(m)$	Critic-induced tempered policy under message condition m
	$\bar{c}_{ij,t}$	Request-time predicted CGDC score stored with a delayed message
	$\alpha^{\mathrm{cgdc}}_{ij,t}$	CAMA attention weight on message from sender j to receiver i

B EXTENDED RELATED WORK

Communication learning is a longstanding theme in MARL. Most existing approaches study how agents communicate in delay-free settings, but differ in whether they primarily emphasize broad information sharing, selective communication, or delayed-message handling. Below, we summarize these lines of work and clarify the positioning of CDCMA.

B.1 CENTRALIZED COMMUNICATION

Centralized communication protocols enable broad information sharing, commonly through differentiable message passing. Canonical examples include CommNet (Sukhbaatar et al., 2016), DIAL (Foerster et al., 2016), and BiCNet (Peng et al., 2017). Follow-up work improves message expressiveness through recurrent aggregation and richer communication memory, e.g., FCMNet (Wang & Sartoretti, 2022), and enhances discreteness, interpretability, or targeted addressing, e.g., Diff Discrete (Freed et al., 2020), TarMAC (Das et al., 2019), and IS (Kim et al., 2020). Other methods reduce redundancy or improve robustness by filtering or dropping messages, e.g., DCC-MD (Kim et al., 2019b) and $\mathfrak{R}$-MACRL (Xue et al., 2022b), or by introducing auxiliary structures such as

centralized proxies and self-supervised objectives, e.g., IMAC (Wang et al., 2020) and MASIA (Guan et al., 2022). Symbolic or grounded communication has also been explored, e.g., GCL (Mordatch & Abbeel, 2018) and AE-Comm (Lin et al., 2021). While these methods substantially enrich communication modeling in MARL, they are mainly developed for settings in which exchanged messages are consumed within the current decision step and do not explicitly address cross-timestep staleness.

B.2 Selective Communication

Selective communication improves scalability and efficiency by restricting exchange to a subset of agents, entities, or timesteps. Early work studies learned communication structures and interaction graphs, e.g., NeurComm (Chu et al., 2020) and Agent-Entity Graph (Agarwal et al., 2020), as well as bandwidth- or event-constrained communication, e.g., SchedNet (Kim et al., 2019a) and ETCNet (Hu et al., 2023). Other methods introduce utility-driven selection objectives, e.g., I2C (Ding et al., 2020), VBC (Zhang et al., 2019), NDQ (Wang et al., 2019), and TMC (Zhang et al., 2020). More recent work estimates message importance using game-theoretic or counterfactual signals, e.g., SMS (Xue et al., 2022a) and CMVC (Gao et al., 2025), and personalizes communication to specific teammates, e.g., MAIC (Yuan et al., 2022). A broad family of attention- or graph-based selectors and aggregators has also been developed, including ATOC (Jiang & Lu, 2018), Gated-ACML (Mao et al., 2020), G2ANet (Liu et al., 2020), MAGIC (Niu et al., 2021), AERL (Pu et al., 2022), InforMARL (Nayak et al., 2023), FlowComm (Du et al., 2021), TGCNet (Zhang et al., 2025), and GRDC (Gao et al., 2026). Recent extensions further consider adversarial robustness, hierarchical dependency modeling, and improved aggregation or coordination strategies, e.g., AME (Sun et al., 2023), DHCG (Liu et al., 2023), T2MAC (Sun et al., 2024), and SeqComm (Ding et al., 2024). These methods significantly improve communication efficiency and selectivity, but are mostly designed for delay-free communication or do not explicitly formulate the receiver-side utility of stale messages under cross-timestep delays.

B.3 Communication with Delays

A smaller but closely related line of work explicitly studies MARL communication under delayed message arrivals. DACOM (Yuan et al., 2023) mitigates latency by adjusting waiting times through a delay-aware communication mechanism and is primarily geared toward short-horizon delay handling. CoDe (Song et al., 2025) infers long-term intents through future-action prediction and fuses asynchronous messages via dual alignment of intent and timeliness. These methods highlight the importance of delayed communication, but they do not explicitly cast the problem as evaluating the receiver-side utility of a stale message at the time it is consumed.

Beyond communication delays, RDC (Fu et al., 2025) formulates stochastic individual observation delays and compensates delayed observation components. Its delay acts on locally received observations, whereas our setting concerns learned messages transmitted between agents across decision timesteps.

In contrast, CDCMA is built around an explicit gain–cost view of delayed-message utility under cross-timestep delays. We formalize this setting as DeComm-POMG, where delayed delivery is part of the decision process, and evaluate delayed communication by balancing communication gain against delay cost. DCOS, OTG, and CAMA instantiate this view through partner selection, future-aware outgoing messages, and gain–cost-guided aggregation, respectively.

C Additional Theory for Section 3

C.1 Proof of Proposition 3.2

Proof. Let $\boldsymbol{u}_t = (\boldsymbol{a}_t, \boldsymbol{m}_t^s)$ denote the joint environment–communication action. Given the current augmented state $\widetilde{s}_t = (s_t, \mathcal{B}_t)$ and $\boldsymbol{u}_t$, the next environment state is sampled from $\mathcal{P}(\cdot \mid s_t, \boldsymbol{a}_t)$. The delay variables are sampled from the exogenous Markov delay process φ, and the next message buffer is determined by the current buffer, the outgoing messages, and the sampled delays:

$$\mathcal{B}_{t+1} = F_{\mathcal{B}}\big(\mathcal{B}_t, \boldsymbol{m}_t^s, \boldsymbol{d}_t\big). \tag{15}$$

Therefore, the conditional distribution of $\widetilde{s}_{t+1} = (s_{t+1}, \mathcal{B}_{t+1})$ depends only on $(\widetilde{s}_t, \boldsymbol{u}_t)$, which establishes the controlled Markov property.

The augmented POMG uses the same local observation process as DeComm-POMG and exposes to each receiver the messages delivered by $\mathcal{B}_t$. Under the same decentralized policy and the same realizations of environment transitions and communication delays, the two formulations consequently generate identical local observation, action, and message histories. □

C.2 Proof of Uniform Bound on $V^g_{ij,t}$

Proposition C.1 (Uniform bound on $V^g_{ij,t}$)**.** *Assume the per-timestep reward is bounded as* $|r_i(\widetilde{s}_t, \boldsymbol{a}_t)| \le R_{\max}$ *for all* t *and* $\gamma \in (0,1)$*. Then for any* $(i, j, t, \widetilde{s}_t, \boldsymbol{a}_t)$*,*

$$\left|V^g_{ij,t}(\widetilde{s}_t, \boldsymbol{a}_t)\right| \;\le\; \frac{2R_{\max}}{1-\gamma}. \tag{16}$$

Proof. Fix any augmented state–action pair $(\widetilde{s}_t, \boldsymbol{a}_t)$. By definition, $V^g_{ij,t}(\widetilde{s}_t, \boldsymbol{a}_t) = Q_i^{(+j)}(\widetilde{s}_t, \boldsymbol{a}_t) - Q_i^{(0)}(\widetilde{s}_t, \boldsymbol{a}_t)$. By the triangle inequality,

$$\left|V^g_{ij,t}(\widetilde{s}_t, \boldsymbol{a}_t)\right| \le \left|Q_i^{(+j)}(\widetilde{s}_t, \boldsymbol{a}_t)\right| + \left|Q_i^{(0)}(\widetilde{s}_t, \boldsymbol{a}_t)\right|. \tag{17}$$

For any joint policy $\boldsymbol{\pi}$, the augmented-state action-value satisfies $Q_i^{\boldsymbol{\pi}}(\widetilde{s}_t, \boldsymbol{a}_t) = \mathbb{E}_{\boldsymbol{\pi}}[\sum_{k\ge 0}\gamma^k r_i(\widetilde{s}_{t+k}, \boldsymbol{a}_{t+k}) \mid \widetilde{s}_t, \boldsymbol{a}_t]$. Using $|r_i| \le R_{\max}$ and $|\mathbb{E}[X]| \le \mathbb{E}[|X|]$ gives $\left|Q_i^{\boldsymbol{\pi}}(\widetilde{s}_t, \boldsymbol{a}_t)\right| \le \sum_{k\ge 0}\gamma^k R_{\max} = R_{\max}/(1-\gamma)$. Applying this bound to both $\boldsymbol{\pi}^{(+j)}$ and $\boldsymbol{\pi}^{(0)}$ and substituting into Eq. (17) yields Eq. (16). □

C.3 Proof of Theorem 3.3

Proof. Let $\boldsymbol{\pi}^{\text{timely}}$ and $\boldsymbol{\pi}^{\text{delayed}}$ be defined as in Theorem 3.3. We work on the augmented Markov game induced by their shared message-generation protocol, with any internal state of that protocol included in s_t for analysis; its remaining control variable is the joint environment action $\boldsymbol{a}_t$.

Define the action-value function under $\boldsymbol{\pi}^{\text{delayed}}$ as

$$Q_i^{\text{delayed}}(\widetilde{s}_t, \boldsymbol{a}_t) := \mathbb{E}\left[\sum_{k\ge 0}\gamma^k r_i(\widetilde{s}_{t+k}, \boldsymbol{a}_{t+k}) \,\middle|\, \widetilde{s}_t, \boldsymbol{a}_t, \boldsymbol{\pi}^{\text{delayed}}\right], \tag{18}$$

and the corresponding value function as

$$V_i^{\text{delayed}}(\widetilde{s}_t) := \mathbb{E}_{\boldsymbol{a}_t\sim\boldsymbol{\pi}^{\text{delayed}}(\cdot|\widetilde{s}_t)}\left[Q_i^{\text{delayed}}(\widetilde{s}_t, \boldsymbol{a}_t)\right]. \tag{19}$$

By a standard telescoping argument on the augmented process, we have

$$J_i(\boldsymbol{\pi}^{\text{timely}}) - J_i(\boldsymbol{\pi}^{\text{delayed}}) = \mathbb{E}_{\tau\sim\boldsymbol{\pi}^{\text{timely}}}\Bigg[\sum_{t\ge 0}\gamma^t\Big(\mathbb{E}_{\boldsymbol{a}_t\sim\mu_t}\big[r_i(\widetilde{s}_t, \boldsymbol{a}_t) + \gamma V_i^{\text{delayed}}(\widetilde{s}_{t+1})\big] - \mathbb{E}_{\boldsymbol{a}_t\sim\nu_t}\big[r_i(\widetilde{s}_t, \boldsymbol{a}_t) + \gamma V_i^{\text{delayed}}(\widetilde{s}_{t+1})\big]\Big)\Bigg], \tag{20}$$

where $\mu_t := \boldsymbol{\pi}^{\text{timely}}(\cdot \mid \widetilde{s}_t)$ and $\nu_t := \boldsymbol{\pi}^{\text{delayed}}(\cdot \mid \widetilde{s}_t)$.

The reward bound implies $|V_i^{\text{delayed}}(\widetilde{s})| \le R_{\max}/(1-\gamma)$. For a fixed $\widetilde{s}_t$, define $g_i(\boldsymbol{a}_t) := r_i(\widetilde{s}_t, \boldsymbol{a}_t) + \gamma\,\mathbb{E}[V_i^{\text{delayed}}(\widetilde{s}_{t+1}) \mid \widetilde{s}_t, \boldsymbol{a}_t]$. Then $\|g_i\|_\infty \le R_{\max}/(1-\gamma)$, and the standard total-variation inequality for bounded functions gives

$$|\mathbb{E}_{\mu_t}[g_i] - \mathbb{E}_{\nu_t}[g_i]| \le \frac{2R_{\max}}{1-\gamma}\|\mu_t - \nu_t\|_{\text{TV}}. \tag{21}$$

Substituting this bound yields

$$J_i(\boldsymbol{\pi}^{\text{timely}}) - J_i(\boldsymbol{\pi}^{\text{delayed}}) \le \frac{2R_{\max}}{1-\gamma}\,\mathbb{E}_{\tau\sim\boldsymbol{\pi}^{\text{timely}}}\left[\sum_{t\ge 0}\gamma^t\|\mu_t - \nu_t\|_{\text{TV}}\right]. \tag{22}$$

Moreover, the two policies share the same message-generation protocol and the same environment-action policies for all agents other than i. Under the factorized decentralized joint policy, they differ only in whether agent i conditions its environment action on $m^{\mathrm{tar}}_{ji,t}$ or $m^{r}_{ij,t}$. Hence,

$$\|\mu_t - \nu_t\|_{\mathrm{TV}} \le \left\|\pi_i(\cdot \mid x_{i,t}, m^{\mathrm{tar}}_{ji,t}) - \pi_i(\cdot \mid x_{i,t}, m^{r}_{ij,t})\right\|_{\mathrm{TV}}. \tag{23}$$

Finally, Pinsker's inequality gives

$$\left\|\pi_i(\cdot \mid x_{i,t}, m^{\mathrm{tar}}_{ji,t}) - \pi_i(\cdot \mid x_{i,t}, m^{r}_{ij,t})\right\|_{\mathrm{TV}} \le \sqrt{\tfrac{1}{2}\,\Delta_{\mathrm{info}}(i,j,t)}. \tag{24}$$

Substituting Eq. (24) into Eq. (22) and taking the positive part of the left-hand side proves Eq. (4). □

D Additional Method Details for Section 4

D.1 Justification for the Critic-Induced Surrogate

During centralized training, we construct the delay-cost surrogate $V^{c}_{ij,t}$ from the tempered policy induced by the gain critic in Eq. (5). Corollary 3.4 relates this surrogate to the actor-policy return-loss bound through explicit actor–probe discrepancy terms. The offline diagnostic in Appendix E.7 compares method-specific decision distributions under complete timely-reference and delayed-message collections.

Proof of Corollary 3.4. By the triangle inequality for total variation,

$$\begin{aligned}\|\pi^{\mathrm{tar}}_i - \pi^{\mathrm{del}}_i\|_{\mathrm{TV}} &\le \|\pi^{\mathrm{tar}}_i - \widetilde{\pi}^{\mathrm{tar}}_i\|_{\mathrm{TV}} + \|\widetilde{\pi}^{\mathrm{tar}}_i - \widetilde{\pi}^{\mathrm{del}}_i\|_{\mathrm{TV}} + \|\widetilde{\pi}^{\mathrm{del}}_i - \pi^{\mathrm{del}}_i\|_{\mathrm{TV}} \\ &\le \delta^{\mathrm{tar}}_{ij,t} + \sqrt{\tfrac{1}{2}V^{c}_{ij,t}} + \delta^{\mathrm{del}}_{ij,t},\end{aligned} \tag{25}$$

where the second inequality follows from Pinsker's inequality and Eq. (6). Substituting Eq. (25) for the actor-policy TV term in Eq. (22) and taking the positive part of the return gap proves Eq. (8). □

Interpretation of the critic-tempered discrepancy. The surrogate measures the change in the gain critic's tempered action distribution between the timely-reference and delayed-message conditions under the same base inputs. It supplies the delay-cost term in CGDC.

D.2 Prior-Regularized Attention View for CAMA

Fix agent i and timestep t with a nonempty available-sender set $\mathcal{Z}_{i,t}$, and define the content score $s_{ij,t} \triangleq q_{i,t}^{\top} k_{ij,t}$. The score stored with the message is $\bar{c}_{ij,t} = \hat{c}_{ij,\kappa_{ji}(t)} > 0$, where $\kappa_{ji}(t)$ is its send timestep; it need not equal the current prediction $\hat{c}_{ij,t}$. These stored scores induce the prior

$$\bar{\alpha}_{ij,t} \triangleq \frac{\bar{c}_{ij,t}}{\sum_{l\in\mathcal{Z}_{i,t}} \bar{c}_{il,t}}, \qquad j \in \mathcal{Z}_{i,t}. \tag{26}$$

Note that $\bar{\alpha}_{ij,t} > 0$ for all $j \in \mathcal{Z}_{i,t}$, hence $\mathrm{KL}(\boldsymbol{\alpha}_t \| \bar{\boldsymbol{\alpha}}_t)$ is well-defined on $\Delta_{\mathcal{Z}_{i,t}}$.

Consider the following KL-regularized objective over the simplex $\Delta_{\mathcal{Z}_{i,t}} = \{\boldsymbol{\alpha}_t : \alpha_{ij,t} \ge 0,\ \sum_{j\in\mathcal{Z}_{i,t}} \alpha_{ij,t} = 1\}$:

$$\mathcal{L}_{\mathrm{attn}}(\boldsymbol{\alpha}_t) = \sum_{j\in\mathcal{Z}_{i,t}} \alpha_{ij,t}\, s_{ij,t} - \frac{1}{\beta}\,\mathrm{KL}\big(\boldsymbol{\alpha}_t \,\|\, \bar{\boldsymbol{\alpha}}_t\big), \tag{27}$$

where $\beta > 0$ and $\mathrm{KL}(\boldsymbol{\alpha}\|\bar{\boldsymbol{\alpha}}) = \sum_{j\in\mathcal{Z}_{i,t}} \alpha_{ij,t} \log \frac{\alpha_{ij,t}}{\bar{\alpha}_{ij,t}}$.

Proposition D.1 (Optimality of CAMA weights)**.** *The unique maximizer of $\mathcal{L}_{\mathrm{attn}}(\boldsymbol{\alpha}_t)$ over $\Delta_{\mathcal{Z}_{i,t}}$ is exactly the CAMA weight $\boldsymbol{\alpha}^{\mathrm{cgdc}}_t = (\alpha^{\mathrm{cgdc}}_{ij,t})_{j\in\mathcal{Z}_{i,t}}$ defined in Eq.* (14)*.*

Proof. For brevity, drop the indices (i,t) and write $\mathcal{Z}$ for $\mathcal{Z}_{i,t}$, s_j for $s_{ij,t}$, α_j for $\alpha_{ij,t}$, and $\bar{\alpha}_j$ for $\bar{\alpha}_{ij,t}$. Consider the Lagrangian with multiplier λ for the constraint $\sum_{j\in\mathcal{Z}} \alpha_j = 1$:

$$\mathcal{J}(\boldsymbol{\alpha}, \lambda) = \sum_{j\in\mathcal{Z}} \alpha_j s_j - \frac{1}{\beta}\sum_{j\in\mathcal{Z}} \alpha_j \log\frac{\alpha_j}{\bar{\alpha}_j} + \lambda\Big(\sum_{j\in\mathcal{Z}} \alpha_j - 1\Big). \tag{28}$$

Taking derivatives and setting them to zero gives, for each $j \in \mathcal{Z}$, $0 = \frac{\partial \mathcal{J}}{\partial \alpha_j} = s_j - \frac{1}{\beta}\big(\log \frac{\alpha_j}{\bar{\alpha}_j} + 1\big) + \lambda$. Rearranging yields $\alpha_j = C\,\bar{\alpha}_j \exp(\beta s_j)$, where $C > 0$ is independent of j. Enforcing $\sum_{j\in\mathcal{Z}} \alpha_j = 1$ gives $\alpha_j = \bar{\alpha}_j \exp(\beta s_j) / \sum_{l\in\mathcal{Z}} \bar{\alpha}_l \exp(\beta s_l)$. Substituting Eq. (26) recovers Eq. (14). Uniqueness follows because $-\mathrm{KL}(\cdot\|\bar{\boldsymbol{\alpha}})$ is strictly concave on the simplex and the first term is linear. □

D.3 Training Scheme

CDCMA comprises seven components: observation encoder e_i, DCOS predictor ϕ_i, OTG predictors (f_i^a, f_i^o), CAMA projections (W_i^Q, W_i^K, W_i^V), decentralized actor π_i, a centralized full critic Q_i^{full}, and an auxiliary gain critic Q_i^{gain}. We share network parameters across agents and condition networks on an agent-identity embedding when needed.

Under centralized training with decentralized execution, we train an off-policy actor–critic with target networks $(\cdot)^-$. For each agent i, the full critic is trained by minimizing the TD loss

$$\mathcal{L}(\theta^{Q_i^{\text{full}}}) = \mathbb{E}\big[(Q_i^{\text{full}}(\boldsymbol{o}, \boldsymbol{a}, \widetilde{m}_i) - y_i)^2\big], \tag{29}$$

where $y_i = r_i + \gamma\, Q_i^{\text{full}-}(\boldsymbol{o}', \boldsymbol{a}', \widetilde{m}_i')$ and $\boldsymbol{a}'$ is formed by sampling each $a_i' \sim \pi_i^-(e_i', \widetilde{m}_i')$.

Gain critic for critic-induced surrogates. To instantiate the critic-induced tempered policy in Eq. (5) and the gain surrogate in Eq. (9), we introduce an auxiliary gain critic Q_i^{gain}. For each ordered pair (i, j) with $j \neq i$, we evaluate a present/absent message pair under the same base inputs $(a_i, \boldsymbol{o}_{-j}, \boldsymbol{a}_{-ij})$, namely $Q_i^{\text{gain}}(a_i, \boldsymbol{o}_{-j}, \boldsymbol{a}_{-ij}, m_{ij}^r)$ and $Q_i^{\text{gain}}(a_i, \boldsymbol{o}_{-j}, \boldsymbol{a}_{-ij}, m^{\emptyset})$. The two targets share the sampled reward and continuation actions but use next-step critic inputs with and without the message, respectively.

Concretely, the gain critic is trained by minimizing

$$\begin{aligned}\mathcal{L}(\theta^{Q_i^{\text{gain}}}) = \mathbb{E}\Bigg[\sum_{j\in\mathcal{N}\setminus\{i\}} \Big(&\big(Q_i^{\text{gain}}(a_i, \boldsymbol{o}_{-j}, \boldsymbol{a}_{-ij}, m_{ij}^r) - y_{ij}^{\text{pres}}\big)^2 \\ &+ \big(Q_i^{\text{gain}}(a_i, \boldsymbol{o}_{-j}, \boldsymbol{a}_{-ij}, m^{\emptyset}) - y_{ij}^{\text{abs}}\big)^2\Big)\Bigg].\end{aligned} \tag{30}$$

where the TD targets are $y_{ij}^{\text{pres}} = r_i + \gamma\, Q_i^{\text{gain}-}(a_i', \boldsymbol{o}_{-j}', \boldsymbol{a}_{-ij}', m_{ij}^{r\,'})$ and $y_{ij}^{\text{abs}} = r_i + \gamma\, Q_i^{\text{gain}-}(a_i', \boldsymbol{o}_{-j}', \boldsymbol{a}_{-ij}', m^{\emptyset})$. Here $m_{ij}^{r\,'}$ denotes the received message from j to i in the next-timestep transition. We sample $\boldsymbol{a}'$ once from the target actors conditioned on the full received-message aggregates, $a_i' \sim \pi_i^-(e_i', \widetilde{m}_i')$, and reuse it in both targets.

Interpretation of the gain surrogate. Both branches reuse the sampled reward and continuation actions, while differing in their current and bootstrap message inputs. Their value difference supplies a message-conditioning signal for CGDC, motivated by the policy-level gain in Eq. (1). The policy-level quantity compares distinct continuation policies, whereas the implemented surrogate compares learned value predictions under shared continuation actions.

Actor–critic updates. For each actor update, we use local inputs from replay and resample $a_i \sim \pi_i(\cdot \mid e_i, \widetilde{m}_i)$ from the current actor rather than reusing its recorded action. We update the actor using the score-function policy gradient with the centralized full critic:

$$\nabla_{\theta^{\pi_i}} \mathcal{J}(\theta^{\pi_i}) = \mathbb{E}\big[\nabla_{\theta^{\pi_i}} \log \pi_i(a_i \mid e_i, \widetilde{m}_i)\, Q_i^{\text{full}}(\boldsymbol{o}, \boldsymbol{a}, \widetilde{m}_i)\big]. \tag{31}$$

We backpropagate through the observation encoder jointly with the actor:

$$\nabla_{\theta^{e_i}} \mathcal{J}(\theta^{e_i}) = \mathbb{E}\Big[\nabla_{\theta^{e_i}} e_i(o_i)\, \nabla_{e_i} \log \pi_i(a_i \mid e_i, \widetilde{m}_i)\, Q_i^{\text{full}}(\boldsymbol{o}, \boldsymbol{a}, \widetilde{m}_i)\Big]. \tag{32}$$

The CAMA projections (W_i^Q, W_i^K, W_i^V) are optimized jointly with the actor by backpropagating its objective through the aggregated message representation $\widetilde{m}_i$.

OTG objectives. OTG is trained with supervised objectives. The observation-dynamics predictor f_i^o regresses the one-step observation difference $(o_i' - o_i)$ conditioned on the local embedding e_i, the available incoming messages $\{m_{ij}^r\}_{j\in\mathcal{Z}_i}$, and the OTG action prediction $\hat{\boldsymbol{a}}_i^{\,g}$:

$$\mathcal{L}(\theta^{\zeta_i}) = \mathbb{E}\Big[\big((o_i' - o_i) - f_i^o\big(e_i, \{m_{ij}^r\}_{j\in\mathcal{Z}_i}, \hat{a}_i^{\,g}\big)\big)^2\Big], \tag{33}$$

where $\mathcal{Z}_i$ denotes the available-sender set in the sampled transition. In addition, the action predictor f_i^a predicts teammates' discrete actions from the same inputs and is trained by the cross-entropy loss

$$\mathcal{L}(\theta^{\psi_i}) = \mathbb{E}\Big[\sum_{j\in\mathcal{Z}_i}\big(-a_j^\top \log \hat{a}_j\big)\Big]. \tag{34}$$

Pairwise CGDC target construction. For every ordered pair (i, j) with $j \neq i$, centralized training constructs a delayed-message reception condition while holding the replay observations and joint action fixed. Within this target calculation, $m_{ji,k}^s$ denotes the sent message and $m_{ij,t}^r$ its received representation under the constructed condition, with $t = k + d_{ji,k}$. We evaluate both $\widehat{V}_{ij,t}^g$ and $V_{ij,t}^c$ using this same $m_{ij,t}^r$ and the same base critic inputs; the timely reference is $m_{ji,t}^{\mathrm{tar}} = e_{j,t}$. The absence of an actual arrival does not set either term to zero. This construction supplies CGDC labels and does not replace the recorded messages used by DCOS, CAMA, the actor, or the full critic, or change the actual available-sender set $\mathcal{Z}_{i,t}$.

DCOS objective. DCOS predicts these pairwise targets from its local observation and actual received-message context by minimizing

$$\mathcal{L}(\theta^{\phi_i}) = \mathbb{E}\Big[\sum_{j\in\mathcal{N}\setminus\{i\}}(c_{ij} - \hat{c}_{ij})^2\Big]. \tag{35}$$

The targets use the minibatch-wise gain–cost weight defined below.

Weight scaling and numerical boundary cases. For a training minibatch $\mathcal{B}$, let $\mathcal{I}_\mathcal{B} = \{(i, j, t) \in \mathcal{B} \mid j \in \mathcal{Z}_{i,t}\}$ contain the tuples with an available received message. When $\mathcal{I}_\mathcal{B} \neq \varnothing$, define

$$\bar{g}_+ := \frac{1}{|\mathcal{I}_\mathcal{B}|}\sum_{(i,j,t)\in\mathcal{I}_\mathcal{B}}[\widehat{V}_{ij,t}^g]_+, \qquad \bar{v}_c := \frac{1}{|\mathcal{I}_\mathcal{B}|}\sum_{(i,j,t)\in\mathcal{I}_\mathcal{B}} V_{ij,t}^c. \tag{36}$$

With numerical threshold $\epsilon_c = 10^{-8}$, set

$$\lambda_0 := \begin{cases} 0, & \mathcal{I}_\mathcal{B} = \varnothing, \\ 0, & \mathcal{I}_\mathcal{B} \neq \varnothing \text{ and } \bar{v}_c \leq \epsilon_c, \\ \mathrm{stopgrad}(\bar{g}_+/\bar{v}_c), & \text{otherwise}, \end{cases} \qquad \lambda = \kappa\lambda_0, \quad \kappa \geq 0. \tag{37}$$

The default multiplier is $\kappa = 1$. The positive part is applied to each gain only for scale estimation; the target retains the signed gain, $c_{ij,t} = \widehat{V}_{ij,t}^g - \lambda V_{ij,t}^c$. The arrival-based subset above is used only for scale estimation; CGDC labels are constructed for all non-self pairs. The low-cost branch avoids amplifying negligible KL values, and $\mathrm{stopgrad}$ detaches the weight from differentiation. DCOS learns these targets as the critics and minibatch scales evolve. At zero delivery delay, $H = 0$; the same weight rule applies to the computed surrogate cost, which is zero only when the timely and received conditions induce identical probe distributions.

D.4 Time Complexity Analysis

We distinguish decentralized-execution complexity from CGDC target construction during centralized training. Action-space size, per-agent representation dimensions, and network widths are held fixed; dependence on the number of input agents and messages is retained explicitly.

Let $\mathcal{G}_{i,t} = \{j \neq i \mid \hat{c}_{ij,t} > 0\}$ be the senders requested by receiver i, $\mathcal{Z}_{i,t}$ the senders whose messages are available to i, and $\mathcal{S}_t = \bigcup_i \mathcal{G}_{i,t}$ the requested-sender set. Define $b_q := \max_i |\mathcal{G}_{i,t}|$, $b_r := \max_i |\mathcal{Z}_{i,t}|$, and $n_s := |\mathcal{S}_t| \leq N$.

Decentralized-execution complexity. DCOS outputs $N(N-1)$ scores per timestep. Let $C_{\mathrm{D}}(N, b_r)$ denote its per-receiver forward cost and $C_{\mathrm{O}}(b_r)$ the cost of one OTG prediction step, including incoming-message processing and teammate-action prediction. Each requested sender constructs one message for all its receivers, giving $\mathcal{O}(n_s[1 + HC_{\mathrm{O}}(b_r)])$ construction cost. Link delivery and CAMA aggregation cost $\mathcal{O}(N(b_q + b_r))$. The total per-timestep cost is therefore

$$\mathcal{O}(NC_{\mathrm{D}}(N, b_r) + n_s[1 + HC_{\mathrm{O}}(b_r)] + N(b_q + b_r)) .$$

For linear-cost scoring and message processing, $C_{\mathrm{D}} = \mathcal{O}(N + b_r)$ and $C_{\mathrm{O}} = \mathcal{O}(1 + b_r)$, yielding $\mathcal{O}(N^2 + n_s H(1 + b_r))$. This becomes $\mathcal{O}(N^2 + NH)$ when b_r is bounded independently of N, and $\mathcal{O}(N^2(1 + H))$ for dense arrivals.

Additional training-time complexity. The gain and delay-cost surrogates are not evaluated during decentralized execution. Given the constructed pairwise message conditions, evaluating the CGDC targets for B sampled transitions requires $\mathcal{O}(BN^2)$ gain-critic evaluations at fixed action-space size. If one evaluation costs $C_Q(N)$, this critic-evaluation cost is $\mathcal{O}(BN^2C_Q(N))$, excluding message-condition construction. The dependence on N accounts for the other agents' observations and actions in the critic input; for example, direct concatenation with a fixed-width first layer gives $C_Q(N) = \mathcal{O}(N)$. Target construction is performed once after each training episode. For an episode minibatch, B counts all nonpadding transitions in the sampled episodes.

Dependence on team size. Under the linear-processing assumptions above and fixed H, execution has quadratic worst-case dependence on N. Sender-level message reuse avoids repeating OTG rollouts for each requested link, while sparse arrivals reduce the message-processing cost within each rollout.

E Additional Experimental Details and Results

E.1 Implementation Details

Environment details. For Cooperative Navigation and Predator Prey in MPE, we remove teammate vision to strengthen partial observability. Each episode lasts 60 timesteps, and each agent chooses from the discrete action set {up, down, left, right, stop}.

Cooperative Navigation. Eight agents and eight landmarks are randomly placed. The agents aim to occupy as many landmarks as possible while minimizing collisions. Each agent observes only the positions of its three nearest landmarks and has no access to teammate information, requiring agents to infer teammates' intentions when choosing landmarks.

Predator Prey. Eight predators cooperate to capture four faster-moving prey. Each predator observes the relative positions and velocities of its three nearest prey but has no access to teammate information. The prey follow a fixed strategy of moving away from the nearest predator. Predators have size 0.04 and acceleration 5.0, while prey have size 0.05 and acceleration 7.0. The prey's speed advantage makes individual capture difficult and motivates coordinated pursuit.

For SMAC, we use SC2.4.10 with difficulty 7, and each episode lasts 50 timesteps.

Evaluation protocol. Each configuration is trained for 200,000 episodes with four independent random seeds and evaluated every 500 episodes over 32 evaluation episodes. Performance tables use seed-level evaluation means from the final checkpoint; zero-shot tests reuse the `easy`-trained checkpoint without further updates. For SMAC performance in Tables 1 and 5, we report the mean and sample standard deviation ($n - 1$ denominator, $n = 4$) of the unsmoothed final-checkpoint win rates, rounded to two decimal places in percent. Learning curves first apply a seven-point moving average to each seed's evaluation sequence; their means and shaded standard deviations are then computed across the smoothed seed curves.

CDCMA optimization schedule. For CDCMA and its ablations, training starts without a replay-buffer warm-up. When fewer than 32 episodes are available, we sample with replacement to form a 32-episode minibatch. We perform one gradient step after each training episode and update the target networks every 200 gradient steps.

Table 4: Hyperparameters of CDCMA and its ablations.

Hyperparameter	MPE	SMAC
Discount factor	0.96	0.99
Actor learning rate	10^{-3}	10^{-4}
Critic learning rate	10^{-2}	10^{-3}
Actor hidden dimension	64	64
Critic hidden dimension	128	128
Message dimension	64	64
Batch size (episodes)	32	32
Replay-buffer capacity (episodes)	5×10^3	5×10^3
Rollout workers	1	1
Activation function	ReLU	ReLU
Optimizer	Adam	Adam

The ablations are CDCMA-DCOS, CDCMA-CAMA, CDCMA-OTG, and CDCMA-PlainAttn. MPE includes Cooperative Navigation and Predator Prey.

Delayed-message handling protocol. All methods share the same delay distributions and message-delivery rules. Agents may use only messages that arrive before action selection. When multiple messages from the same sender arrive simultaneously at a receiver, we retain only the most recent one and discard older arrivals from that sender at the same timestep. In CDCMA, a receiver's request is modeled as an instantaneous control signal that is not subject to the delayed content channel. Once a request is issued, the sender constructs the outgoing message at the same timestep. The request-time CGDC score is stored in a separate scalar tensor aligned with the corresponding message; the message and score undergo the same sampled delivery delay and are used together by CAMA upon arrival. Each timestep contains one message-construction round. DCOS and OTG use the communication inputs after previously sent messages have arrived and before new messages are sent. The channel delivers newly generated zero-delay messages before CAMA aggregation and action selection; these arrivals do not trigger another construction round. A current zero-delay message replaces any older arrival from the same sender together with its score. For actual-message diagnostics, timely references are stored for the final available-sender set used for action selection, with empty references for silent pairs.

Baseline implementations. We retain the algorithm-specific hyperparameters of the official implementations of DACOM, TGCNet, T2MAC, SMS, G2ANet, and ATOC, without additional tuning. All methods follow the common task settings, training budget, evaluation protocol, and delayed-message interface described above. Hyperparameters for CDCMA and its ablations are listed in Table 4.

CoDe reproduction. We reproduce CoDe (Song et al., 2025) with a QMIX backbone and Adam at a learning rate of 10^{-3}. The agent GRU and mixer have hidden dimensions of 64 and 32, respectively; the two-layer hypernetworks have a hidden dimension of 64. Its timeliness alignment uses the framework-provided message age Δt, which is not an input to CDCMA. The delay interface changes delivery timing while preserving CoDe's message cache and timeliness alignment.

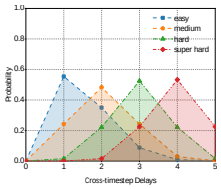
DACOM adaptation. We omit DACOM's TimeNet module for within-step waiting and evaluate the remaining architecture under the common cross-timestep delivery protocol. All reported DACOM results use this variant.

Communication payload. CDCMA uses 64-dimensional messages, with one additional scalar request-time CGDC score stored separately for each selected message. OTG changes message construction but not dimensionality. The official-code baselines retain their native message representations.

CDCMA-specific defaults. Unless otherwise stated, we use $H = d_{\max} = 5$ under delayed communication and $H = 0$ in delay-free evaluations. We use $\eta = 1$ for the critic-tempered policy and the standard scaled dot-product factor $\beta = 1/\sqrt{d_k}$ for CAMA attention, where d_k is the key dimension. Neither value is tuned separately.

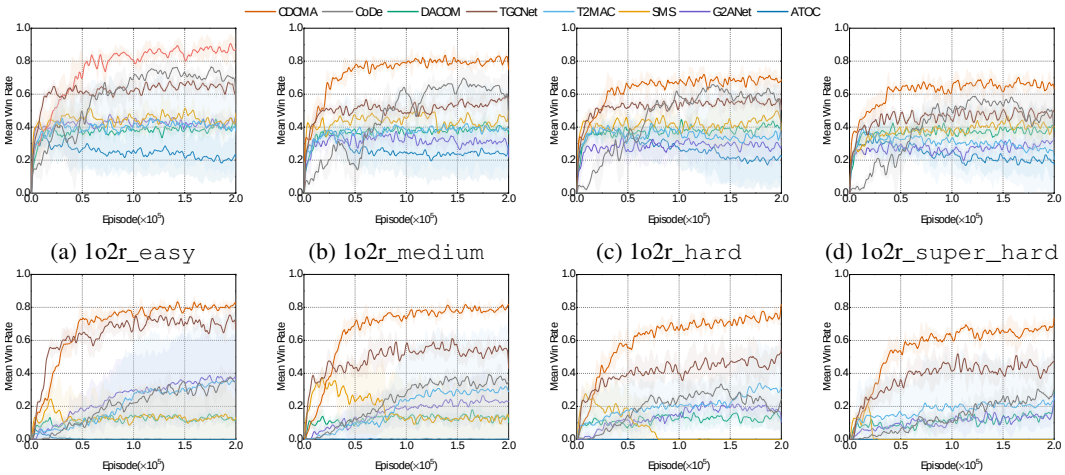
Difficulty ℓ	μ_ℓ	σ_ℓ
`easy`	1.00	1.00
`medium`	2.00	0.80
`hard`	3.00	0.70
`super_hard`	4.00	0.70

(a) Delay parameters (μ_ℓ, σ_ℓ).



(b) Illustration of the four delay distributions.

Figure 5: Delay model: parameter table and corresponding distribution shapes.



(a) 1o2r_easy (b) 1o2r_medium (c) 1o2r_hard (d) 1o2r_super_hard

(e) 1o10b_easy (f) 1o10b_medium (g) 1o10b_hard (h) 1o10b_super_hard

Figure 6: Mean win rate vs. training episodes across two SMAC tasks and four delay difficulties.

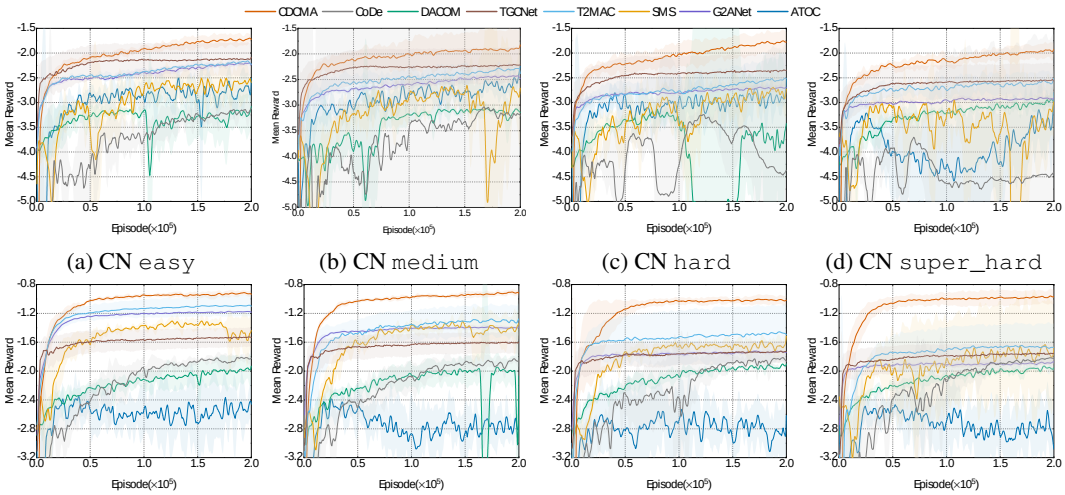
E.2 Delay Model

For each ordered pair (j, i) and each send timestep k, the environment samples an integer delay $d_{ji,k} \in \{1, 2, \ldots, d_{\max}\}$ with $d_{\max} = 5$ in the delayed settings. The delay-free setting corresponds to $d_{ji,k} = 0$ for all ordered pairs. We use a difficulty-indexed probability mass function $p_\ell(d)$ on $\{1, \ldots, d_{\max}\}$ with $\ell \in \{\texttt{easy}, \texttt{medium}, \texttt{hard}, \texttt{super_hard}\}$. Our default specification is a truncated discrete normal:

$$p_\ell(d) = \frac{\Phi\left(\frac{d+0.5-\mu_\ell}{\sigma_\ell}\right) - \Phi\left(\frac{d-0.5-\mu_\ell}{\sigma_\ell}\right)}{\sum_{u=1}^{d_{\max}} \left[\Phi\left(\frac{u+0.5-\mu_\ell}{\sigma_\ell}\right) - \Phi\left(\frac{u-0.5-\mu_\ell}{\sigma_\ell}\right)\right]}, \tag{38}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function. The parameters (μ_ℓ, σ_ℓ) are listed in Fig. 5(a), and the resulting PMF shapes are shown in Fig. 5(b). As the difficulty increases, μ_ℓ shifts probability mass toward longer delays, while σ_ℓ controls the spread.

At the beginning of each timestep k, for each ordered pair (j, i), the environment samples an exogenous per-link delay $d_{ji,k} \sim p_\ell(\cdot)$ independently across ordered pairs and timesteps. Any message generated at timestep k from j to i is delivered after $d_{ji,k}$ timesteps. Message arrival and the rule for concurrent arrivals follow the DeComm-POMG specification in Sec. 3: the arrival timestep is $t = k + d_{ji,k}$, and if several messages from the same sender arrive simultaneously, only the most recent one is retained.



(a) CN `easy` (b) CN `medium` (c) CN `hard` (d) CN `super_hard`

(e) PP `easy` (f) PP `medium` (g) PP `hard` (h) PP `super_hard`

Figure 7: Mean episode reward vs. training episodes on Cooperative Navigation (CN) and Predator Prey (PP) across four delay difficulties.

E.3 Additional Learning Curves

Fig. 6 and Fig. 7 complement Table 1 by showing training curves on SMAC and MPE, respectively. The curves illustrate learning progress across delay settings; smoothing and cross-seed aggregation follow Appendix E.1.

E.4 Delay-Free Evaluation

Fig. 8 reports the delay-free learning curve on `1o_10b_vs_1r`. In the zero-delay limit, CDCMA remains competitive and stable across seeds, indicating that the framework does not rely on delayed settings to remain effective.

E.5 Additional Validation on MMM2

To broaden the empirical scope beyond the two main SMAC maps reported in the main text, we additionally evaluate CDCMA on MMM2 under the same five settings: `delay_free`, `easy`, `medium`, `hard`, and `super_hard`. As shown in Table 5, CDCMA achieves the highest mean win rate across all five settings. This provides an additional evaluation on a third SMAC map.

In the delay-free setting, CDCMA achieves a mean win rate of 89.06%, compared with 84.38% for TGCNet, and remains ahead across the four delayed settings.

E.6 OTG Prediction Diagnostics

We provide qualitative diagnostics for OTG on Predator Prey under the `super_hard` delay setting. In this task, each predator observes its ego position and velocity together with the relative positions and velocities of its three nearest prey. OTG predicts a future observation trajectory $\hat{\tau}_{i,t} = (\hat{o}_{i,t+1}, \ldots, \hat{o}_{i,t+H})$, which is encoded into outgoing messages to reduce the information gap at consumption time.

Fig. 9 shows that OTG can track short-horizon observation evolution reasonably well, which is consistent with the empirical gains obtained by using a moderate prediction horizon. At the same time, we observe occasional prediction-loss spikes. These spikes are typically associated with nearest-neighbor identity switching in the observation encoding: the prey slots are defined by nearest-neighbor ordering at each timestep, so the slot-to-entity correspondence may change between successive steps. As a result, the supervised target for a fixed coordinate slot may correspond to a different physical

Table 5: Additional validation on MMM2. Results are mean win rate (std).

Setting	CDCMA	CoDe	DACOM	TGCNet	T2MAC	SMS	G2ANet	ATOC
delay_free	**89.06**(3.13)	3.13(6.25)	6.25(4.42)	84.38(5.71)	5.47(4.69)	15.63(14.21)	29.69(24.54)	0(0)
easy	**89.06**(3.13)	5.47(6.93)	6.25(7.22)	71.88(10.83)	0.00(0.00)	7.81(13.62)	22.66(15.18)	0(0)
medium	**85.16**(4.69)	4.69(9.38)	5.47(1.56)	69.53(13.35)	0(0)	4.69(5.41)	21.88(11.97)	0(0)
hard	**84.38**(4.42)	10.16(11.80)	3.91(2.99)	61.72(18.11)	0(0)	1.56(1.80)	19.53(18.29)	0(0)
super_hard	**82.03**(5.34)	14.06(14.09)	4.69(5.41)	58.59(9.67)	0(0)	0(0)	0(0)	0(0)

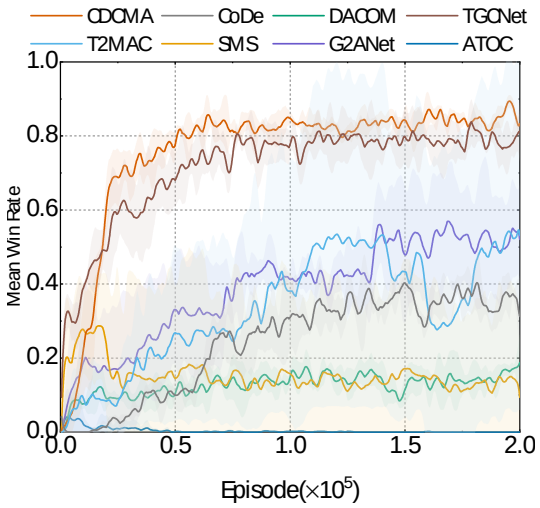
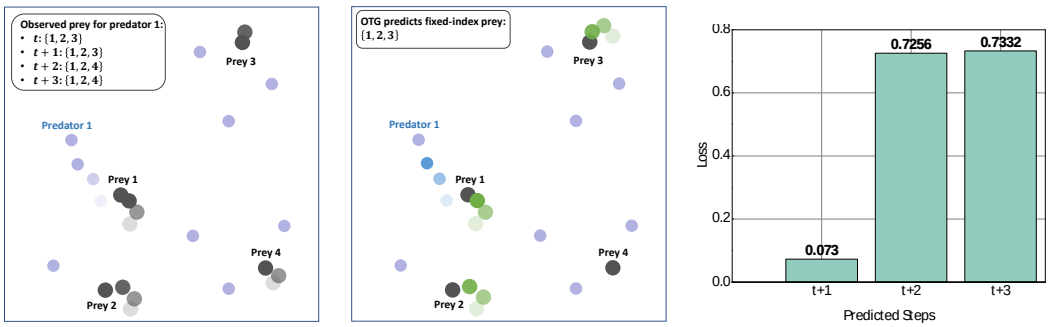


Figure 8: Mean win rate vs. training episodes on 1o_10b_vs_1r under the delay-free condition.



(a) True trajectory (b) Predicted trajectory (c) Prediction loss

Figure 9: Qualitative OTG diagnostics on Predator Prey. Left: ground-truth observation trajectories. Middle: OTG-predicted trajectories. Right: per-timestep prediction loss. Lighter circles indicate farther future steps.

prey in the next step, which can produce a large apparent error even when the predicted physical motion is reasonable.

This permutation-induced mismatch becomes more likely as the prediction horizon grows, which helps explain the degradation observed when $H > d_{\max}$ in the ablation results. Overall, the diagnostics support the interpretation that OTG is beneficial for mitigating temporal misalignment under delays, while overly long horizons amplify both model error and supervision mismatch.

E.7 Additional CGDC Diagnostics

We provide additional diagnostics for CDCMA on PP and 1o_10b_vs_1r under the bounded stochastic delay settings used in the main experiments. We first specify the decision-distribution metric reported in Sec. 5.4, then examine gain and score consistency.

Decision-distribution information gap. At receiver timestep t, define the delayed and timely message collections over the same available-sender set as

$$\mathbf{m}_{i,t}^{r} = (m_{ij,t}^{r})_{j\in\mathcal{Z}_{i,t}}, \qquad \mathbf{m}_{i,t}^{\mathrm{tar}} = (e_{j,t})_{j\in\mathcal{Z}_{i,t}}.$$

We evaluate the same decision network under both collections, holding the receiver's local inputs, history state, and all other inputs fixed. Only the message contents change; aggregation is recomputed using the method's existing aggregation module. For CDCMA, the stored request-time CGDC scores remain fixed.

Both message conditions use the same current available-action set $\mathcal{A}_{i,t}^{\mathrm{avail}}$. For policy-based methods, $p_i(\cdot \mid x_{i,t}, \mathbf{m})$ is the actor's output distribution over this set. For the Q-based T2MAC, let $Q_i^{\mathrm{act}}(x_{i,t}, \mathbf{m}, a)$ denote the learned action values used for decision making, and define the diagnostic distribution as

$$p_i(a \mid x_{i,t}, \mathbf{m}) = \frac{\exp Q_i^{\mathrm{act}}(x_{i,t}, \mathbf{m}, a)}{\sum_{a'\in\mathcal{A}_{i,t}^{\mathrm{avail}}} \exp Q_i^{\mathrm{act}}(x_{i,t}, \mathbf{m}, a')}, \qquad a \in \mathcal{A}_{i,t}^{\mathrm{avail}}.$$

The KL divergence is computed over this same available-action set. T2MAC's diagnostic therefore depends on its native action-value scale; CDCMA's diagnostic uses its actor rather than either centralized critic. We define

$$\Delta_{\mathrm{dec}}(i,t) = D_{\mathrm{KL}}\big(p_i(\cdot \mid x_{i,t}, \mathbf{m}_{i,t}^{\mathrm{tar}}) \,\big\|\, p_i(\cdot \mid x_{i,t}, \mathbf{m}_{i,t}^{r})\big)\,. \tag{39}$$

For N agents and T evaluated timesteps, we report

$$\overline{\Delta}_{\mathrm{dec}} = \frac{1}{NT}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\Delta_{\mathrm{dec}}(i,t). \tag{40}$$

When $\mathcal{Z}_{i,t} = \varnothing$, both message collections are empty, yielding zero KL that remains included in the mean. The overall mean reflects both message availability and the sensitivity of the decision distribution to message replacement. The diagnostic leaves each method's action-selection rule unchanged and uses delayed inputs for environment interaction.

Equation (3) isolates a single sender's effect on the actor policy, whereas Eq. (39) replaces the complete available-message collection and uses the method-specific distribution defined above. CGDC training uses the critic-tempered cost $V_{ij,t}^{c}$ in Eq. (6); Fig. 4 reports the decision-distribution metric defined above.

As shown in Fig. 4, CDCMA has smaller mean decision-distribution gaps than the compared methods across all evaluated delay regimes. From `easy` to `super_hard`, its gap increases from 0.02 to 0.09 on PP and from 0.03 to 0.08 on `1o_10b_vs_1r`. Removing OTG increases the gap on both tasks. Together with the module ablations in Fig. 3a, these results are consistent with OTG's role in reducing timely-vs-delayed changes in the receiver's action distribution.

Gain and score consistency. We examine whether the gain surrogate and current CGDC scores agree with the full critic's assessment of received messages. For each evaluated transition, we consider only receiver–sender pairs (i, j, t) with $j \in \mathcal{Z}_{i,t}$ and compare the original aggregation with a message-masked counterfactual. The message-present condition keeps all received messages, including that from sender j, and yields $\tilde{m}_i^{\mathrm{all}}$, while the message-masked condition replaces $m_{ij,t}^{r}$ with $m^{\emptyset}$ and recomputes aggregation, yielding $\tilde{m}_i^{(-j)}$. With the local observation, joint action, and other received messages fixed, we define

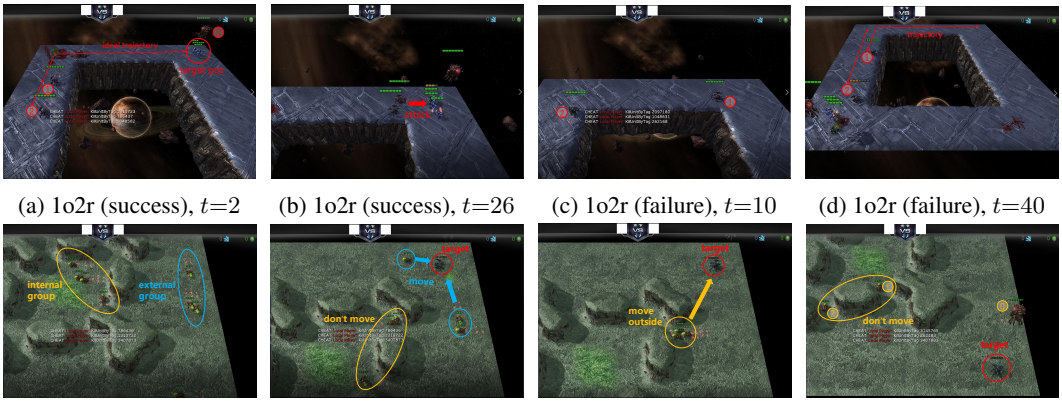
$$\Delta Q_{ij,t} = Q_i^{\mathrm{full}}(\boldsymbol{o}_t, \boldsymbol{a}_t, \tilde{m}_i^{\mathrm{all}}) - Q_i^{\mathrm{full}}(\boldsymbol{o}_t, \boldsymbol{a}_t, \tilde{m}_i^{(-j)}). \tag{41}$$

We report raw ΔQ without normalization; positive values indicate a higher full-critic prediction when the message is retained. This diagnostic uses a leave-one-out comparison with other messages retained, while Eq. (1) compares sender-only access with no incoming messages.

For each seed and delay setting, we compare mean ΔQ between the bottom and top 30% of pairs ranked by $\widehat{V}_{ij,t}^{g}$, and between pairs with current DCOS scores $\hat{c}_{ij,t} \le 0$ and $\hat{c}_{ij,t} > 0$. Sign agreement is the fraction of pairs for which $\hat{c}_{ij,t}$ and raw $\Delta Q_{ij,t}$ have matching signs. We average these statistics equally across the four seeds and four delay settings.

Table 6: CGDC consistency diagnostics. ΔQ denotes the raw full-critic value difference between retaining and masking a received message.

Task	Low-gain ΔQ	High-gain ΔQ	ΔQ ($\hat{c} \leq 0$)	ΔQ ($\hat{c} > 0$)	Sign agreement (%)
PP	-0.019	0.088	-0.013	0.076	79.1
`1o_10b_vs_1r`	-0.012	0.121	-0.018	0.109	82.1



(a) 1o2r (success), t=2 (b) 1o2r (success), t=26 (c) 1o2r (failure), t=10 (d) 1o2r (failure), t=40

(e) 1o10b (success), t=8 (f) 1o10b (success), t=18 (g) 1o10b (success), t=32 (h) 1o10b (failure)

Figure 10: Replay snapshots of CDCMA policies on `1o_2r_vs_4r` and `1o_10b_vs_1r` under cross-timestep delays.

Table 6 shows higher raw ΔQ for high-gain than low-gain pairs and for positive-score than nonpositive-score pairs on both tasks. Sign agreement reaches 79.1% on PP and 82.1% on `1o_10b_vs_1r`. These results indicate consistency between the gain surrogate, current CGDC scores, and the full critic's message assessments.

E.8 Replay Visualizations for Qualitative Analysis

We provide replay snapshots in Fig. 10 to qualitatively examine how CDCMA facilitates coordination under cross-timestep delays on two SMAC scenarios. The top row corresponds to `1o_2r_vs_4r`, whereas the bottom row corresponds to `1o_10b_vs_1r`. For each scenario, we juxtapose representative successful and failed rollouts.

`1o_2r_vs_4r`. Roaches depend on the Overseer for target-related information and use it to synchronize navigation and engagement. In the successful rollout, the two Roaches exhibit consistent movement toward the target region early in the episode, thereby avoiding route divergence around the corner, and later enter effective firing range in a temporally aligned manner. In the failure rollout, an early deviation from the intended route induces split navigation and delays mutual convergence, so the team remains spatially dispersed and fails to complete effective engagement within the episode horizon.

`1o_10b_vs_1r`. Banelings communicate with the Overseer to acquire target-location cues and to coordinate regrouping under terrain constraints. In the successful rollout, agents first separate into external and internal groups because of the cliff structure, then the internal group exits the cliffs and rejoins the main movement toward the target region, restoring team cohesion before engagement. In the failure case, the internal group remains trapped or inactive inside the cliffs while the target stays separated, so the remaining agents fail to re-establish coordinated movement before termination.

Overall, these replay snapshots suggest that successful outcomes are associated with earlier target-aligned movement and more timely regrouping, whereas failures are characterized by persistent route divergence or delayed coordination under stale communication.

F CDCMA ALGORITHM

Algorithm 1 CDCMA

Initialize parameter-shared networks: observation encoder e, actor π, full critic Q^{full}, gain critic Q^{gain}, DCOS predictor ϕ, OTG predictors (f^a, f^o), and CAMA projections (W^Q, W^K, W^V).
Initialize target networks, replay buffer $\mathcal{R}$, and per-pair delay queues $\{\mathcal{B}_{ji}\}_{j\neq i}$.
for episode $= 1, \dots, M$ **do**
  Reset environment and queues; initialize an episode record and observe $\{o_{i,0}\}_{i=1}^N$.
  for $t = 0, \dots, T-1$ **do**
    The environment samples exogenous delays $\{d_{ji,t}\}_{j\neq i}$, unobserved by agents.
   // Deliver arrivals at timestep t
    for agent $i = 1, \dots, N$ **do**
      Deliver all entries whose arrival timestep equals t from $\{\mathcal{B}_{ji}\}_{j\neq i}$.
      For each sender $j \neq i$, set $(m^r_{ij,t}, \bar{c}_{ij,t})$ to the delivered entry with the largest send timestep if any; otherwise set $m^r_{ij,t} \leftarrow m^{\emptyset}$.
      Let $\mathcal{Z}_{i,t} \leftarrow \{\, j \neq i \mid m^r_{ij,t} \neq m^{\emptyset} \,\}$.
    end for
   // Partner selection
    for agent $i = 1, \dots, N$ **do**
      Observe $o_{i,t}$ and received messages $\{m^r_{ij,t}\}_{j\in\mathcal{Z}_{i,t}}$; compute $e_{i,t} \leftarrow e_i(o_{i,t})$.
      Compute $\{\hat{c}_{ij,t}\}_{j\neq i} \leftarrow \phi_i\big(o_{i,t}, \{m^r_{ij,t}\}_{j\in\mathcal{Z}_{i,t}}; \theta^{\phi_i}\big)$ and set $\mathcal{G}_{i,t} \leftarrow \{\, j \neq i \mid \hat{c}_{ij,t} > 0 \,\}$.
    end for
   // Construct sender-level messages
    Let $\mathcal{S}_t \leftarrow \bigcup_{i=1}^N \mathcal{G}_{i,t}$ be the requested-sender set.
    for sender $j \in \mathcal{S}_t$ **do**
      Roll out $\hat{\tau}_{j,t}$ with (f^a, f^o) using pre-send inputs ($H = 0$ when delay-free).
      Encode $(\tau_{j,t}, \hat{\tau}_{j,t})$ into one outgoing message $m^s_{j,t}$.
    end for
   // Deliver or enqueue on requested links
    for receiver $i = 1, \dots, N$ **do**
      for each sender $j \in \mathcal{G}_{i,t}$ **do**
        If $d_{ji,t} = 0$, the channel replaces $(m^r_{ij,t}, \bar{c}_{ij,t})$ with $(m^s_{j,t}, \hat{c}_{ij,t})$ and adds j to $\mathcal{Z}_{i,t}$; otherwise enqueue for $t + d_{ji,t}$.
      end for
    end for
   // Aggregate received messages and act
    for agent $i = 1, \dots, N$ **do**
      If $\mathcal{Z}_{i,t} = \varnothing$, set $\widetilde{m}_{i,t} \leftarrow \mathbf{0}$; otherwise aggregate $\{m^r_{ij,t}\}_{j\in\mathcal{Z}_{i,t}}$ with CGDC-modulated attention.
      Select action $a_{i,t} \sim \pi_i(\cdot \mid e_{i,t}, \widetilde{m}_{i,t})$.
    end for
    Execute joint action $\boldsymbol{a}_t$; each agent observes $o_{i,t+1}$ and receives $r_{i,t}$.
   // Store transition and actual-message references
    For each pair (i, j) with $j \neq i$, set $m^{\text{tar}}_{ji,t} \leftarrow e_{j,t}$ if $j \in \mathcal{Z}_{i,t}$, and $m^{\text{tar}}_{ji,t} \leftarrow m^{\emptyset}$ otherwise.
    Append the transition to the episode record, including observations, actions, rewards, received messages, stored request-time scores, and timely references.
  end for
  Store the completed episode in $\mathcal{R}$.
 // One training update after the episode
  Sample a 32-episode minibatch from $\mathcal{R}$, using replacement if $|\mathcal{R}| < 32$, and evaluate losses on valid transitions.
  For every pair $j \neq i$, construct a delayed-message reception condition as in Appendix D.3, without altering the recorded communication inputs.
  Evaluate $\widehat{V}^g_{ij,t}$ and $V^c_{ij,t}$ on this same condition using Eqs. (9) and (6), with timely reference $e_{j,t}$.
  Compute λ_0 by Eq. (37) on the minibatch; set $\lambda \leftarrow \kappa\lambda_0$ (default $\kappa = 1$).
  For each pair (i, j) with $j \neq i$, compute the CGDC target $c_{ij,t} \leftarrow \widehat{V}^g_{ij,t} - \lambda V^c_{ij,t}$.
  Update the full critic using Eq. (29), the gain critic using Eq. (30), the actor and encoder using Eqs. (31) and (32), CAMA jointly with the actor, OTG using Eqs. (33) and (34), and DCOS using Eq. (35).
  If 200 gradient updates have elapsed since the last target update, update the target networks.
end for