Master's Project

Uncertainty-Guided Selective Communication for Cooperative Multi-Agent Reinforcement Learning

Seongil Heo

April 10, 2026

Computer Science

University of Utah

Committee: Daniel S. Brown; Tucker Hermans; Tom Henderson

Abstract

Cooperative multi-agent reinforcement learning requires agents to coordinate under partial observability, but always-on communication can be costly and unrealistic in systems with bandwidth, latency, or energy constraints. A key challenge is determining when communication is genuinely useful: not merely whether agents can exchange messages, but whether a message improves the current decision. We address this challenge by proposing an uncertainty-guided selective communication framework that uses decision entropy reduction as a trigger for inter-agent communication. Our method introduces a teacher-supervised student gate trained to predict whether incorporating a neighbor's message will sharpen the agent's action distribution. During training, a teacher path compares the no-communication and with-communication policies and uses their entropy difference as a direct supervision signal for the gate. During decentralized execution, the gate is computed from the agent's local observation representation and does not require centralized information. The framework further combines top-k neighbor selection, attention-based message construction, and a residual distributional critic that decomposes the with-communication return distribution into a local base component and a communication-induced residual. We evaluate the approach in POGEMA, a partially observable multi-agent pathfinding benchmark, across Random, Mazes, and Warehouse maps. The learned gate stabilizes at activation rates that increase with environment complexity, approximately 0.25–0.35 on Random maps and 0.5–0.6 on Warehouse maps, suggesting that the framework adapts communication frequency to the coordination demands of the environment. Compared with always-on and random communication baselines, our method achieves a better cost–utility trade-off on structured maps, and its uncertainty-based gating criterion generalizes to unseen map geometries without environment-specific recalibration.

Contents

1 Introduction

Cooperative multi-agent reinforcement learning (MARL) is an important framework for coordination problems in multi-robot navigation, warehouse automation, and autonomous systems. However, coordination becomes difficult under partial observability, where each agent observes only a limited part of the environment and must act without full knowledge of other agents' states, intentions, or future actions. This missing information can lead to inefficient coordination, redundant behavior, and collisions.

Communication can reduce the information gap in cooperative MARL by allowing agents to exchange useful information. Prior work has proposed differentiable communication protocols and attention-based message passing [1, 2], and later methods introduced gating and targeting mechanisms to make communication more selective [3, 4]. However, many of these methods either assume that communication is always available or do not explicitly decide when communication is necessary.

Always-on communication is often inefficient and unrealistic in practical multi-agent systems. In real-world robotic systems, communication may be limited by bandwidth, latency, interference, energy constraints, or hardware limitations. This has motivated recent work on communication-efficient and selective communication in MARL [5].

We propose an uncertainty-guided selective communication framework for cooperative MARL. Our method uses a teacher-supervised student gate to decide when communication should influence the policy. During training, a teacher signal measures whether message fusion reduces decision uncertainty, while during execution the student gate predicts communication usefulness from local observations alone. The method combines local neighbor selection, attention-based message construction, gated message fusion, and a residual distributional critic that separates the no-communication value from the communication-induced value change.

We evaluate our approach in POGEMA, a grid-based multi-agent pathfinding benchmark under partial observability. We compare against no-communication, always-communication, random-communication, and attention-based communication baselines. The results show that selective communication improves coordination while reducing unnecessary message usage, providing a favorable trade-off between task performance and communication efficiency.

Our contributions are summarized as follows:

1. We propose a selective communication framework for cooperative MARL under partial observability, where agents learn when to use communication instead of assuming always-on message exchange.
2. We introduce a teacher-supervised student gate that learns communication usefulness from local observations, using uncertainty reduction from message fusion as a training signal.
3. We combine local neighbor selection, attention-based message construction, gated message fusion, and a residual distributional critic to support efficient communication and value decomposition.

The remainder of this paper is organized as follows. Section 2 provides background on cooperative MARL, CTDE, and distributional value estimation. Sections 3 and 4 review related work and formulate the selective communication problem. Section 5 presents the proposed method, Sections 6 and 7 describe the experiments and results, and Section 8 concludes the report.

2 Background

2.1 Cooperative Multi-Agent Reinforcement Learning

Cooperative multi-agent reinforcement learning (MARL) studies decision-making problems in which multiple agents interact with a shared environment and work together to optimize a common objective [6]. Unlike single-agent reinforcement learning, where one agent selects actions to maximize its own return, cooperative MARL requires multiple agents to coordinate their behavior. This coordination is challenging because each agent's action can affect not only its own future observations and rewards, but also the outcomes of other agents.

A standard way to model cooperative MARL under partial observability is a decentralized partially observable Markov decision process (Dec-POMDP) [7], defined as

$$\mathcal{G} = \langle \mathcal{S},\, \{\mathcal{A}_i\}_{i=1}^{N},\, \{\mathcal{O}_i\}_{i=1}^{N},\, P,\, r,\, \gamma \rangle, \tag{1}$$

where N is the number of agents, $\mathcal{S}$ is the global state space, $\mathcal{A}_i$ and $\mathcal{O}_i$ are the action and observation spaces of agent i, P is the transition function, r is the shared reward function, and $\gamma \in [0, 1)$ is the discount factor.

At each timestep t, the environment is in a global state $s_t \in \mathcal{S}$. Each agent i receives a local observation $o_i^t \in \mathcal{O}_i$, which provides only partial information about the global state, and selects an action $a_i^t \in \mathcal{A}_i$. The joint action $\mathbf{a}^t = (a_1^t, \ldots, a_N^t)$ induces a transition $P(s_{t+1} \mid s_t, \mathbf{a}^t)$ and a shared team reward $r_t = r(s_t, \mathbf{a}^t)$. The objective is to learn decentralized policies $\pi_i(a_i^t \mid o_i^t)$ that maximize the expected discounted team return

$$J(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{T} \gamma^t r_t\right]. \tag{2}$$

In partially observable environments, each agent must make decisions using incomplete information. This can lead to coordination failures when important information about other agents, obstacles, goals, or future interactions is not available from local observations alone. Communication is one way to reduce this information gap, but deciding how and when to communicate remains a key challenge.

2.2 Centralized Training and Decentralized Execution

A common paradigm in cooperative MARL is centralized training with decentralized execution (CTDE) [8, 9]. Under CTDE, agents are trained using additional centralized information—such as the global state or joint observations—but during execution each agent makes decisions using only its own local observation.

This paradigm is useful because training with centralized information reduces learning instability caused by non-stationarity: from the perspective of an individual agent, the environment appears non-stationary because the policies of other agents change during training. A centralized critic can exploit global information to provide more stable learning signals while keeping each agent's policy decentralized.

In actor-critic methods, this is implemented by pairing decentralized actors with a centralized critic. Each agent maintains a policy $\pi_i(a_i^t \mid o_i^t)$, while the critic may condition on the global state s_t or the joint observa-

tion $\mathbf{o}^t$:

$$V(s_t) \quad \text{or} \quad Q(s_t, \mathbf{a}^t). \tag{3}$$

The actor is used at execution time; the critic is used only during training to estimate returns and compute policy gradients.

In this work, we follow the CTDE setting. This allows value estimation to use centralized information during training while keeping the learned policy compatible with decentralized execution—an important distinction because our goal is to improve decentralized coordination under partial observability, not to assume global state access at execution time.

2.3 Distributional Value Estimation

Most actor-critic methods estimate a scalar value function representing the expected return:

$$V(s_t) = \mathbb{E}\left[\sum_{k=t}^{T} \gamma^{k-t} r_k\right]. \tag{4}$$

This scalar summarizes the mean outcome but discards all information about the variability of possible returns. Two states may share the same expected value yet differ substantially in outcome uncertainty.

Distributional reinforcement learning addresses this limitation by modeling the full return distribution [10, 11]. Rather than a scalar, a distributional critic estimates a random variable

$$Z(s_t) = \sum_{k=t}^{T} \gamma^{k-t} r_k, \qquad V(s_t) = \mathbb{E}[Z(s_t)]. \tag{5}$$

In practice, $Z(s_t)$ is represented by K quantile predictions $\{z_1(s_t), \ldots, z_K(s_t)\}$, and the expected value is approximated as

$$V(s_t) \approx \frac{1}{K} \sum_{k=1}^{K} z_k(s_t). \tag{6}$$

The spread of the predicted quantiles serves as a measure of value uncertainty. A simple range-based measure is

$$u(s_t) = z_K(s_t) - z_1(s_t), \tag{7}$$

while a mean-deviation measure is

$$u(s_t) = \frac{1}{K} \sum_{k=1}^{K} \left| z_k(s_t) - \bar{z}(s_t) \right|, \qquad \bar{z}(s_t) = \frac{1}{K} \sum_{k=1}^{K} z_k(s_t). \tag{8}$$

A larger spread indicates that the critic predicts a wider range of possible returns, which can be interpreted as higher value uncertainty. In partially observable multi-agent environments, such uncertainty often arises when an agent lacks sufficient information about other agents' states or intentions. This connection between quantile spread and information gap motivates our use of distributional value estimation as a signal for when communication is useful, which we develop in Section 5.

3 Related Work

3.1 Communication-Efficient Multi-Agent Reinforcement Learning

Communication has long been recognized as an important mechanism for improving coordination under partial observability in cooperative MARL. Early methods established two distinct lines of work: differentiable communication, where agents learn continuous message protocols jointly with their policies [1, 2], and selective communication, where agents additionally learn when or to whom to send messages [3, 4]. While these methods demonstrated that learned communication can substantially improve cooperative performance, they generally assume that communication channels are reliable and always available, which may be unrealistic in physical multi-agent systems.

Motivated by practical bandwidth, latency, and energy constraints, recent work has shifted from enabling communication to reducing its cost while preserving cooperative performance. Proposed approaches include pruning messages that are unlikely to be beneficial [5], generating compact consensus representations to reduce redundant information exchange [12], and building spatiotemporal information hubs that amortize communication over time [13]. Our work follows this direction but asks a different question: rather than reducing the number of messages through fixed structural choices, we ask whether an agent's uncertainty about its own decision can serve as a principled trigger for communication.

3.2 Selective and Context-Aware Communication

Beyond simple bandwidth reduction, a growing body of work treats communication itself as a learned, adaptive decision. Instead of broadcasting at every timestep, these methods aim to determine when to communicate, which agents to communicate with, or what information should be exchanged. IC3Net [3] learns a binary gate per agent from task rewards, allowing agents to suppress communication entirely when it is not profitable. Attention-based communication methods such as TarMAC [4] learn to focus on relevant teammates when constructing messages, while more recent communication-efficient methods reduce redundant information exchange through compact shared representations or structured communication pathways [12, 13].

These methods are closely related to our motivation because they treat communication as adaptive rather than fixed. However, the criterion for communication is typically a learned context embedding, an attention score, or a message-utility estimate derived from task rewards. These signals provide only indirect feedback about whether communication actually improved the agent's decision. In contrast, our work uses decision uncertainty reduction as an explicit and interpretable signal: we measure whether incorporating a message reduces the entropy of the agent's action distribution, and train the communication gate to predict this reduction from local observations alone.

3.3 Uncertainty Estimation in MARL and Distributional Reinforcement Learning

Distributional reinforcement learning models the full distribution of returns rather than only its expectation, as introduced by Bellemare et al. [10] and Dabney et al. [11] and discussed further in Section 2.3. The spread of the predicted quantile distribution provides a natural measure of value uncertainty, and recent work has shown that this signal carries information beyond the expected value [14].

In MARL, uncertainty arises from multiple sources: partial observability, stochastic environment dynamics, and the non-stationarity induced by other agents' evolving policies. Prior work has exploited uncertainty estimates primarily for exploration and safety. For instance, Hazra et al. [15] integrate distributional critics with safety constraints in cooperative settings, and Wen et al. [16] use uncertainty-aware world models for connected autonomous vehicles.

Our work differs from these approaches in how uncertainty is used. We do not use it to shape exploration or penalize risk. Instead, we use the reduction in decision uncertainty caused by message fusion as a training signal for a communication gate. This connects the agent's decision confidence directly to its communication behavior: an agent that is already confident from its local observation has less reason to use information from neighbors, whereas an uncertain agent may benefit from incorporating messages. This uncertainty-driven criterion is complementary to the bandwidth- and structure-oriented approaches reviewed above, and it is different from reward-based gating or attention-only communication.

4 Problem Formulation

We formulate selective communication in cooperative MARL under partial observability. The goal is to learn decentralized policies that achieve high cooperative performance while communicating only when communication is genuinely useful. This section defines the decision problem at an abstract level; the architecture that implements the communication gate is described in Section 5.

4.1 Cooperative Partially Observable Setting

We adopt the Dec-POMDP framework introduced in Section 2.1, with N cooperative agents sharing a team reward r_t. Each agent i receives a local observation o_i^t at timestep t and selects an action a_i^t; the joint action $\mathbf{a}^t = (a_1^t, \ldots, a_N^t)$ determines the next state and shared reward. The objective is to maximize the expected discounted team return

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t r_t \right]. \tag{9}$$

Because each agent observes only a limited region of the environment, local observations frequently lack the information needed for optimal coordination. An agent may be unaware of other agents' locations, intentions, or planned trajectories. This partial observability can produce decision uncertainty that manifests as redundant actions, inefficient paths, or inter-agent collisions.

4.2 Selective Communication

To reduce the information gap caused by partial observability, agents may exchange messages with neighbors. Let M_i^t denote the aggregated message available to agent i at timestep t, constructed from messages sent by nearby agents (the exact construction is specified in Section 5).

We introduce a binary communication gate for each agent:

$$g_i^t \in \{0, 1\}. \tag{10}$$

The gate determines whether the incoming message should influence the agent's policy. Concretely, the

communication-aware policy is

$$\pi_i\big(a_i^t \mid o_i^t,\ g_i^t \cdot M_i^t\big) \ = \ \begin{cases} \pi_i(a_i^t \mid o_i^t) & \text{if } g_i^t = 0, \\ \pi_i(a_i^t \mid o_i^t, M_i^t) & \text{if } g_i^t = 1. \end{cases} \tag{11}$$

When $g_i^t = 0$, the agent acts using only its local observation; when $g_i^t = 1$, it incorporates the incoming message. This formulation treats communication as a learned decision rather than an always-available channel. Each agent must determine both which action to take and whether external information is useful for that decision.

4.3 Communication-Efficient Objective

The core trade-off is between cooperative task performance and communication cost. We express this with the augmented objective

$$J(\pi, g) = \mathbb{E}_{\pi,g}\left[\sum_{t=0}^{T} \gamma^t \left(r_t - \lambda \sum_{i=1}^{N} g_i^t \right)\right], \tag{12}$$

where $\lambda \geq 0$ is a communication penalty coefficient.

Setting $\lambda = 0$ removes the explicit penalty; communication efficiency is then measured post-hoc via the empirical communication rate $\frac{1}{NL}\sum_{t,i} g_i^t$. Setting $\lambda > 0$ directly encourages agents to communicate only when the expected return gain exceeds the cost, yielding a principled cost–utility trade-off.

The problem is therefore to jointly learn decentralized policies $\{\pi_i\}_{i=1}^N$ and communication gates $\{g_i\}_{i=1}^N$ that maximize $J(\pi, g)$. The central challenge is that the gate must remain decentralized: at execution time, g_i^t must be computed from o_i^t alone, without access to other agents' observations or future actions. We address this challenge through the teacher-supervised student gate described in Section 5.

5 Method

We propose an uncertainty-guided selective communication framework for cooperative MARL. The method addresses two questions: which agents should be considered as communication candidates, and when the received message should influence the policy. As illustrated in Figure 1, the architecture consists of three components: (i) an actor-side communication module that constructs and gates messages, (ii) a train-only teacher supervision path that provides uncertainty-based gate targets, and (iii) a residual distributional critic that decomposes the value contribution of communication.

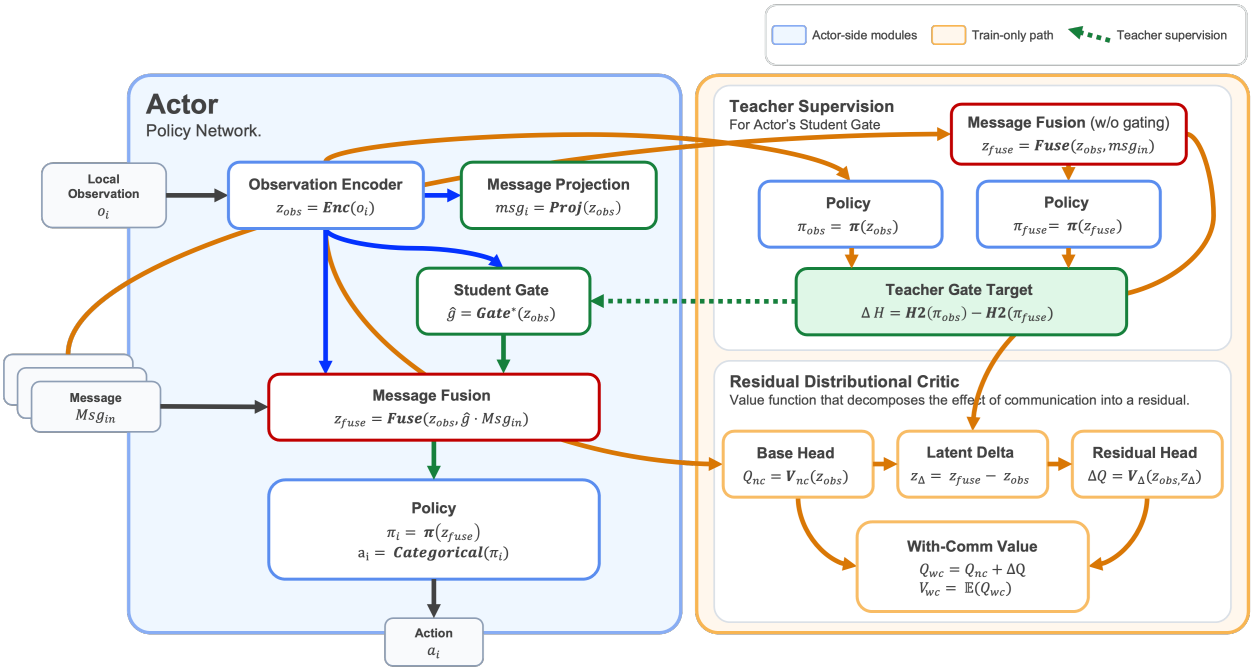


Figure 1: Overview of the proposed uncertainty-guided selective communication architecture. Each agent encodes its local observation, constructs messages from selected neighbors, applies a student communication gate, and fuses useful communication into the policy representation. During training, a teacher path supervises the gate using uncertainty reduction, while a residual distributional critic decomposes the value of communication.

5.1 Overview

Figure 1 summarizes the full pipeline. Each agent i encodes its local observation o_i^t into a latent representation $z_{\text{obs},i}^t$, which is projected into a draft message msg_i^t. A subset of nearby agents' draft messages is aggregated via attention into an incoming message Msg_i^t. A student gate $\hat{g}_i^t$, computed solely from $z_{\text{obs},i}^t$, controls whether Msg_i^t is fused into the fused representation $z_{\text{fuse},i}^t$, from which the policy π_θ samples an action. During training, a teacher path computes gate targets from uncertainty reduction; the critic decomposes the with-communication value into a local base and a communication-induced residual. At execution time the teacher path and critic are removed. Details follow in Sections 5.2–5.8.

5.2 Neighbor Selection

Communicating with all agents introduces unnecessary overhead and noisy information. Each agent therefore selects a local neighborhood before message aggregation. Let $d(i, j)$ denote the distance between agents i and j. The candidate set within communication radius R_{comm} is

$$\mathcal{C}_i^t = \{j \neq i : d(i, j) \leq R_{\text{comm}}\}, \tag{13}$$

from which we retain the k nearest agents:

$$\mathcal{N}_i^t = \text{TopK}_{j \in \mathcal{C}_i^t}(-d(i, j),\ k). \tag{14}$$

Only agents in $\mathcal{N}_i^t$ contribute messages, focusing aggregation on locally relevant neighbors.

5.3 Message Construction with Attention

Each agent i projects its observation representation into a draft message and a query vector:

$$msg_i^t = W_{msg}\, z_{\mathrm{obs},i}^t, \qquad q_i^t = W_q\, z_{\mathrm{obs},i}^t. \tag{15}$$

For each neighbor $j \in \mathcal{N}_i^t$, the key and value vectors are

$$k_j^t = W_k\, msg_j^t, \qquad v_j^t = W_v\, msg_j^t. \tag{16}$$

Scaled dot-product attention weights are

$$\alpha_{ij}^t = \frac{\exp\big((q_i^t)^\top k_j^t/\sqrt{d}\big)}{\sum_{\ell\in\mathcal{N}_i^t}\exp\big((q_i^t)^\top k_\ell^t/\sqrt{d}\big)}, \tag{17}$$

where d is the key dimension. The aggregated incoming message is

$$Msg_i^t = \sum_{j\in\mathcal{N}_i^t} \alpha_{ij}^t\, v_j^t. \tag{18}$$

This module determines which neighbors are most relevant to agent i's current decision.

5.4 Student Communication Gate and Message Fusion

The student gate decides whether Msg_i^t should influence the policy. It is computed solely from the local representation to preserve decentralization:

$$\hat{g}_i^t = \sigma\big(G_\phi(z_{\mathrm{obs},i}^t)\big) \in [0,1], \tag{19}$$

where G_ϕ is the gate network and σ is the sigmoid function.

Training (soft gate). During training, $\hat{g}_i^t$ is used as a continuous weight. This provides a continuous communication weight during training and allows gradients to flow through the gate.

$$z_{\mathrm{fuse},i}^t = \mathrm{Fuse}\big(z_{\mathrm{obs},i}^t,\ \hat{g}_i^t \cdot Msg_i^t\big), \tag{20}$$

Execution (hard gate). At test time, $\hat{g}_i^t$ is thresholded to obtain a binary decision:

$$g_i^t = \mathbb{I}[\hat{g}_i^t > \tau_g], \tag{21}$$

and the fused representation becomes

$$z_{\mathrm{fuse},i}^t = \mathrm{Fuse}(z_{\mathrm{obs},i}^t,\ g_i^t \cdot Msg_i^t). \tag{22}$$

When $g_i^t = 0$, the message input is removed; when $g_i^t = 1$, the full incoming message is fused into the policy representation. This corresponds to the case-based policy in Eq. (11).

In both phases the policy outputs an action distribution from $z_{\text{fuse},i}^t$:

$$\pi_i^t = \pi_\theta(\cdot \mid z_{\text{fuse},i}^t), \qquad a_i^t \sim \pi_i^t. \tag{23}$$

5.5 Teacher Supervision for the Student Gate

The student gate must make a communication decision from local information alone. To provide a training signal, we use a train-only teacher path that computes the uncertainty reduction caused by message fusion.

No-communication and with-communication policies. We compute two policies sharing the same network π_θ but with different inputs:

$$\pi_{\text{obs},i}^t = \pi_\theta(\cdot \mid z_{\text{obs},i}^t), \tag{24}$$

$$z_{\text{full},i}^t = \text{Fuse}(z_{\text{obs},i}^t,\ Msg_i^t), \tag{25}$$

$$\pi_{\text{fuse},i}^t = \pi_\theta(\cdot \mid z_{\text{full},i}^t). \tag{26}$$

Note that $z_{\text{full},i}^t$ always fuses the full incoming message without applying the gate, so the teacher measures the benefit of using the full message under the current policy architecture.

Uncertainty reduction signal. We measure the reduction in decision uncertainty via the top-2 entropy difference:

$$\Delta H_i^t = H_2(\pi_{\text{obs},i}^t) - H_2(\pi_{\text{fuse},i}^t), \tag{27}$$

where

$$H_2(\pi) = -\sum_{a \in \text{Top2}(\pi)} \tilde{\pi}(a) \log \tilde{\pi}(a), \quad \tilde{\pi}(a) = \frac{\pi(a)}{\sum_{b \in \text{Top2}(\pi)} \pi(b)}. \tag{28}$$

We restrict entropy to the top-2 actions because the gate only needs to detect whether the agent is uncertain between its two most likely choices, rather than resolving uncertainty over the full action space. This makes the signal more sensitive to cases where a message changes a near-tie decision, which are the cases where communication is most valuable. A positive ΔH_i^t indicates that message fusion sharpened the decision.

Teacher targets. We convert ΔH_i^t into a gate target. A binary target,

$$y_i^t = \mathbb{I}[\Delta H_i^t > \epsilon], \tag{29}$$

labels timesteps where communication materially reduces uncertainty (margin ϵ). A soft target,

$$y_i^t = \sigma(\Delta H_i^t / \eta), \tag{30}$$

provides a continuous supervision signal calibrated by temperature η. The gate loss is

$$\mathcal{L}_{\text{gate}} = \frac{1}{NT} \sum_{t=0}^{T} \sum_{i=1}^{N} \ell_{\text{gate}}(\hat{g}_i^t,\ y_i^t), \tag{31}$$

where ℓ_{gate} is binary cross-entropy for binary targets or MSE for soft targets.

5.6 Residual Distributional Critic

To make the value contribution of communication explicit, we adopt a residual distributional critic that decomposes the with-communication return into a local base component and a communication-induced residual.

Base head. The base head estimates the no-communication return distribution from the local representation:

$$Q_{\text{nc},i}^t = V_{\text{nc}}(z_{\text{obs},i}^t), \tag{32}$$

represented as K quantiles.

Residual head. The residual critic uses the fused actor representation $z_{\text{fuse},i}^t$ from the gated message fusion module. We define the communication-induced latent delta as

$$z_{\Delta,i}^t = z_{\text{fuse},i}^t - z_{\text{obs},i}^t, \tag{33}$$

and the residual head predicts the value change due to communication:

$$\Delta Q_i^t = V_\Delta(z_{\text{obs},i}^t,\ z_{\Delta,i}^t). \tag{34}$$

With-communication value. The combined distributional estimate is

$$Q_{\text{wc},i}^t = Q_{\text{nc},i}^t + \Delta Q_i^t, \qquad V_{\text{wc},i}^t = \mathbb{E}[Q_{\text{wc},i}^t]. \tag{35}$$

The base head captures what the agent can infer from local observations alone, while the residual head captures the additional value attributable to the communication-induced change in the policy representation. When the gate suppresses communication, $z_{\Delta,i}^t$ becomes small; when communication changes the fused representation, the residual head estimates the corresponding value change.

5.7 Training Objective

The full training loss combines four terms:

$$\mathcal{L} = \mathcal{L}_{\text{actor}} + \beta_v\, \mathcal{L}_{\text{critic}} + \beta_g\, \mathcal{L}_{\text{gate}} + \beta_c\, \mathcal{L}_{\text{comm}} - \beta_e\, \mathcal{H}(\pi). \tag{36}$$

Actor loss. The actor optimizes the communication-penalized return (Eq. (12)) via policy gradient. The penalty coefficient λ provides a task-level incentive to avoid unnecessary communication.

Critic loss. The critic is trained with a quantile regression loss [11]. Let y^t_{ret} denote the distributional return target and $Q^t_{\text{wc},i,k}$ the k-th predicted quantile. Then

$$\mathcal{L}_{\text{critic}} = \frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} \rho_{\tau_k} \big(y^t_{\text{ret}} - Q^t_{\text{wc},i,k}\big), \tag{37}$$

where ρ_{τ_k} is the quantile Huber loss at level τ_k.

Gate loss. $\mathcal{L}_{\text{gate}}$ is the teacher-supervised loss from Eq. (31).

Communication regularizer. $\mathcal{L}_{\text{comm}}$ directly penalizes the mean soft gate value:

$$\mathcal{L}_{\text{comm}} = \frac{1}{NT} \sum_{t=0}^{T} \sum_{i=1}^{N} \hat{g}^t_i. \tag{38}$$

This term discourages the gate from staying always-on during training, complementing the task-level λ penalty in the actor: λ acts on realized (hard) communication decisions via rewards, while $\beta_c \mathcal{L}_{\text{comm}}$ acts directly on the soft gate weight to shape its distribution. When $\lambda = 0$ and $\beta_c = 0$ no explicit communication penalty is applied, and efficiency is evaluated solely through the empirical communication rate.

Entropy bonus. $-\beta_e \mathcal{H}(\pi)$ is a standard entropy regularization term that encourages policy exploration.

5.8 Decentralized Execution

At execution time, the teacher supervision path and the distributional critic are removed. Each agent i executes the following steps using only local information and messages from selected neighbors:

1. Encode: $z^t_{\text{obs},i} = \text{Enc}(o^t_i)$.
2. Select neighbors $\mathcal{N}^t_i$ by radius and top-k.
3. Construct messages and aggregate: Msg^t_i via Eq. (18).
4. Compute gate: $\hat{g}^t_i$ via Eq. (19), threshold to g^t_i via Eq. (21).
5. Fuse and act: $a^t_i \sim \pi_\theta(\cdot \mid z^t_{\text{fuse},i})$.

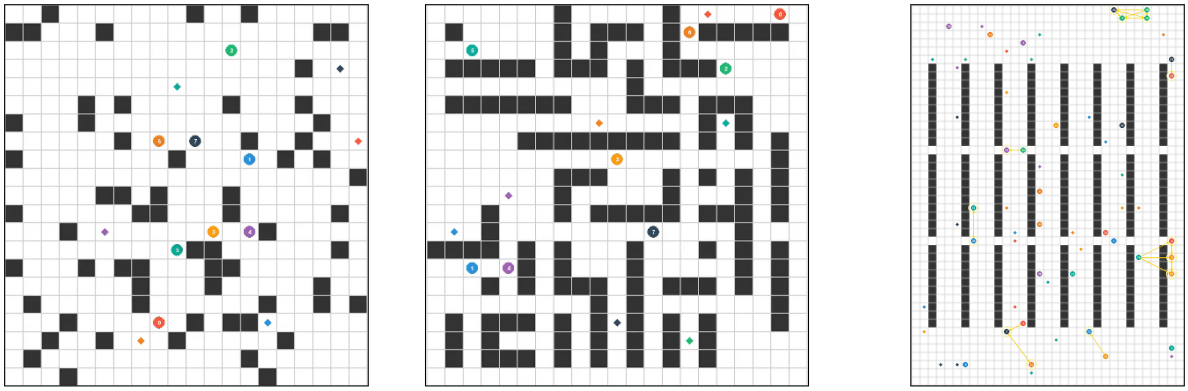
No centralized information is required. The final policy is fully decentralized while benefiting from teacher supervision and residual value decomposition applied during training.

6 Experiments

We evaluate the proposed method in partially observable multi-agent pathfinding environments. The experiments are designed to test whether selective communication improves coordination and whether it achieves a better trade-off between task performance and communication cost than always-on or random communication.

6.1 Environment

We use POGEMA [17], a partially observable grid-based multi-agent pathfinding environment. Each agent is assigned an individual goal and must navigate to it while avoiding obstacles and other agents. The task is cooperative: team performance depends on all agents reaching their goals efficiently.



(a) Random (b) Mazes (c) Warehouse

Figure 2: Example POGEMA scenarios. Random maps test general navigation and local collision avoidance; Mazes test coordination through narrow passages; Warehouse maps create dense agent interactions in structured corridors, where communication is expected to be more useful.

Each agent receives a local observation centered on its current position, including partial information about nearby obstacles, nearby agents, and its own goal. Agents cannot observe the full map and are therefore unaware of other agents' intentions or planned trajectories, making POGEMA a suitable testbed for evaluating communication under partial observability.

We train and evaluate on three map types illustrated in Figure 2. **Random** maps contain randomly placed obstacles and primarily test general navigation and local collision avoidance. **Mazes** contain narrow passages that require careful coordination to avoid deadlocks. **Warehouse** maps feature structured corridor layouts that create dense multi-agent interactions, where communication can be useful for avoiding congestion.

6.2 Baselines

We compare against six baselines covering the spectrum from no communication to always-on communication, and including representative prior methods.

- **No communication.** A MAPPO [9] policy trained with local observations only. This baseline establishes the performance floor achievable without any inter-agent information.
- **Always communication.** Agents always fuse the attention-aggregated message Msg_i^t into the policy, regardless of uncertainty. This provides an upper-reference baseline for using the shared communication module without any gate.
- **Random communication.** The gate g_i^t is sampled randomly at each timestep, providing a baseline that uses communication without conditioning on observations or decision uncertainty.
- **Nearest-neighbor communication.** Agents fuse messages from the top-k nearest neighbors uncondition-

ally (same neighbor selection and attention as our method, but without the teacher-supervised gate). This isolates the contribution of the gate from the contribution of structured neighbor selection and attention.

- **IC3Net** [3]. A gated communication method that learns binary communication decisions end-to-end from task rewards. Unlike our method, the gate receives no explicit uncertainty-based supervision.
- **TarMAC** [4]. An attention-based targeted communication method that learns which agents to attend to when constructing messages. TarMAC addresses which agents to communicate with, but it does not explicitly decide whether communication should be used for the current decision.

The first four baselines share the same network architecture as our method and differ only in the communication strategy, enabling a controlled ablation. IC3Net and TarMAC are re-implemented in the same POGEMA environment to ensure a fair comparison.

6.3 Metrics

We report the following six metrics. Let N be the number of agents, L the episode length, and t_i the timestep at which agent i first reaches its goal.

Success rate. Proportion of agents that reach their goals:

$$\mathrm{SR} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[\text{agent } i \text{ reached goal}]. \tag{39}$$

Episode success. Whether all agents reach their goals in an episode:

$$\mathrm{ES} = \mathbf{1}[\text{all agents reached goal}]. \tag{40}$$

We report ES averaged over evaluation episodes.

Sum of costs (SOC). Total completion time summed over all agents:

$$\mathrm{SOC} = \sum_{i=1}^{N} \begin{cases} t_i, & \text{if agent } i \text{ reached goal,} \\ L, & \text{otherwise.} \end{cases} \tag{41}$$

Lower SOC indicates more efficient team navigation.

Makespan. Completion step of the last successful agent:

$$\mathrm{Makespan} = \begin{cases} \max_{i:\,\text{succeeded}} t_i, & \text{if any agent reached goal,} \\ L, & \text{otherwise.} \end{cases} \tag{42}$$

Lower makespan indicates faster completion among successful agents.

Occurred collisions. Total number of collisions during an episode:

$$\text{Collisions} = \sum_{t=1}^{L} \text{collision\_count}_t, \tag{43}$$

where collision_count_t is the number of collisions occurring at timestep t.

Communication cost (CommCost). Mean fraction of timesteps at which an agent activates its gate:

$$\text{CommCost} = \frac{1}{NL} \sum_{t=1}^{L} \sum_{i=1}^{N} g_i^t, \tag{44}$$

where $g_i^t \in \{0, 1\}$ is the binary gate. CommCost allows us to distinguish methods that improve performance by communicating more from those that achieve a better cost–utility trade-off.

6.4 Evaluation Protocol

We organize the results into five analyses, corresponding to Sections 7.1–7.5:

1. **Uncertainty behavior** (Section 7.1). We track value uncertainty and gate activation rate during training to examine whether low local uncertainty implies that communication is unnecessary.
2. **Effect of communication strategy** (Section 7.2). We compare all methods across map types on SR, ES, SOC, Makespan, and Collisions.
3. **Communication cost–utility trade-off** (Section 7.3). We plot task performance against CommCost for each method to evaluate whether selective communication dominates always-on or random communication in the cost–performance space.
4. **Comparison with existing methods** (Section 7.4). We compare against IC3Net and TarMAC to assess whether teacher-supervised uncertainty-based gating outperforms reward-based and attention-based alternatives.
5. **Generalization and scalability** (Section 7.5). We test policies on map types not seen during training (out-of-distribution generalization) and on team sizes ranging from 8 to 64 agents.

7 Results and Analysis

We analyze the results along five dimensions: uncertainty behavior, the effect of communication strategy, the communication cost–utility trade-off, comparison with existing methods, and robustness under out-of-distribution conditions and larger team sizes. The goal is not only to report task performance, but also to understand when communication is useful and why uncertainty-guided selective communication can outperform always-on and random alternatives.

7.1 Uncertainty Behavior

(a) Random (b) Mazes (c) Warehouse

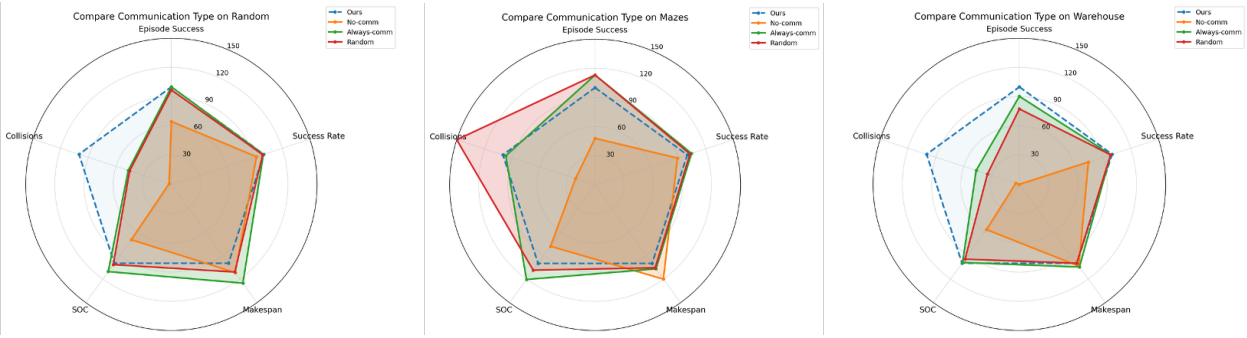
Figure 3: Uncertainty reduction ΔH_i^t (top row) and gate activation rate (bottom row) during training across map types. Uncertainty reduction peaks early and converges toward zero, yet the gate stabilizes at a positive activation rate that varies by map complexity: approximately 0.25–0.35 on Random, 0.3–0.4 on Mazes, and 0.5–0.6 on Warehouse.

Figure 3 shows how value uncertainty and gate activation evolve over 2M training steps. In all three map types, the uncertainty reduction signal ΔH_i^t peaks sharply in the early phase of training (before 250k steps) and then decays toward near-zero as the policy and value networks internalize recurring local patterns. This early peak reflects the initial period during which message fusion provides the most new information, before the encoder learns to anticipate common situations from the local observation alone.

Despite this convergence, the gate activation rate does not collapse to zero. Instead, it stabilizes at a level that reflects the coordination complexity of each map: approximately 0.25–0.35 on Random maps, 0.3–0.4 on Mazes, and 0.5–0.6 on Warehouse maps. This ordering matches the structural complexity of the three environments. Warehouse corridors create more frequent near-tie decisions where neighbor information can change the preferred action, requiring communication more often.

The dissociation between near-zero raw uncertainty and positive gate activation supports the design of the teacher-supervised gate. The gate targets the reduction in decision entropy caused by message fusion rather than the absolute local uncertainty level. An agent whose encoder is confident about its local observation can still benefit from knowing that a neighbor is about to occupy the same corridor, which is the kind of information that ΔH_i^t is designed to capture.

7.2 Effect of Communication Strategy



(a) Random (b) Mazes (c) Warehouse

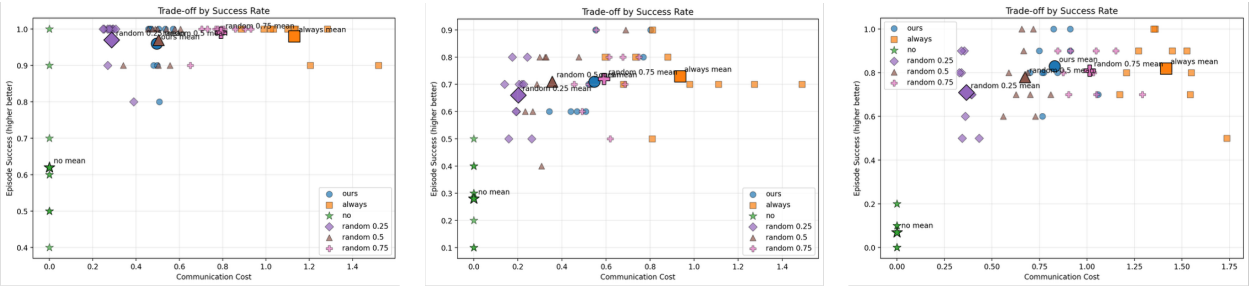
Figure 4: Radar charts comparing four communication strategies (Ours, No-comm, Always-comm, Random-comm) across Random, Mazes, and Warehouse maps on five metrics. Each axis is oriented so that a larger polygon area corresponds to better overall performance. Our method (blue dashed) achieves the largest or near-largest area in all three environments.

Figure 4 compares all four communication strategies across the three map types. In Random maps, all methods with communication achieve similar Success Rate and Episode Success, as the sparse obstacle layout leaves little room for communication to differentiate performance. The gap appears in SOC and Makespan, where our method produces more efficient paths than Always-comm. This suggests that constant message exposure can make agents slightly over-cautious.

The distinction is more pronounced in Mazes. Random-comm produces the highest Collision count among communicating methods, which appears as the smallest Collisions-axis value in the Mazes radar chart. This suggests that randomly timed messages can introduce unhelpful information at critical narrow-passage decision points. Our method avoids this issue by activating the gate selectively, achieving collision counts comparable to Always-comm while improving SOC and Makespan.

Warehouse maps show the strongest communication effect. No-comm falls substantially behind all communicating methods in Success Rate and Episode Success, confirming that dense corridor interactions genuinely require inter-agent information. Our method achieves the largest radar area among all four strategies on Warehouse. It matches Always-comm on success-related metrics while producing lower SOC, indicating that selective communication avoids the over-cautious waiting behavior that can increase completion time under constant message exposure.

7.3 Communication Cost vs. Utility



(a) Random (b) Mazes (c) Warehouse

Figure 5: Communication cost–utility scatter plots (CommCost on x-axis, Episode Success on y-axis) for all methods across three map types. Large markers denote per-method means. Our method (blue circle) achieves Episode Success comparable to Always-comm (orange square) at a substantially lower CommCost.

Figure 5 plots Episode Success against CommCost for each method across all evaluation episodes. Several observations stand out.

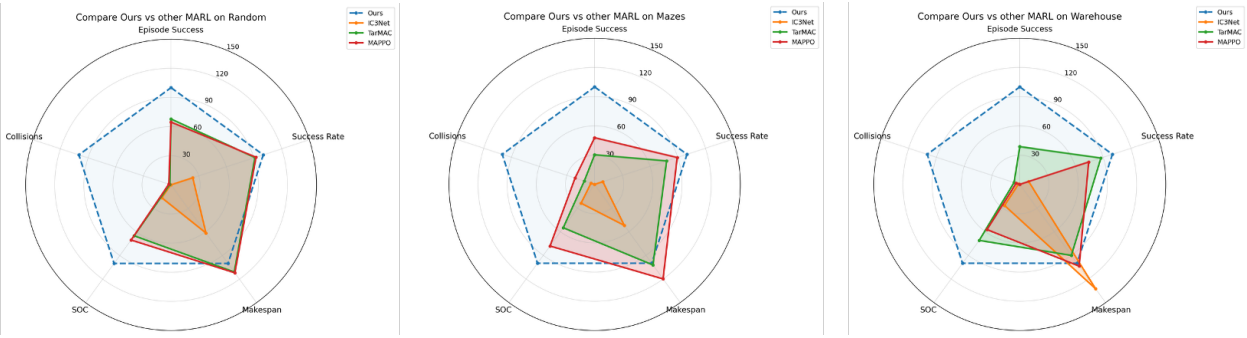
First, No-comm (green star, CommCost $= 0$) achieves high Episode Success on Random but drops markedly on Mazes and Warehouse, illustrating that communication value is environment-dependent rather than universally beneficial.

Second, Always-comm (orange square) achieves high Episode Success in all environments but sits at a CommCost of approximately 1.2–1.5, the highest among all methods. The wide spread of individual episode points indicates high variance, suggesting that always-on communication is sensitive to the specific episode layout.

Third, Random-comm at three gate rates (0.25, 0.5, 0.75) traces a roughly linear frontier: higher random gate rates yield higher CommCost but not consistently higher Episode Success, especially on Mazes where the per-episode variance is large. This suggests that the timing of communication matters as much as its frequency.

Our method (blue circle) clusters near the upper-left region of each scatter plot: its mean CommCost lies below that of Always-comm (approximately 0.4–0.6 depending on the map) while its mean Episode Success is comparable or superior. This is most pronounced in Warehouse, where our mean point sits above the Always-comm mean on the y-axis while remaining to its left on the x-axis. This indicates a Pareto improvement in the cost–utility space.

7.4 Comparison with Existing Methods



(a) Random (b) Mazes (c) Warehouse

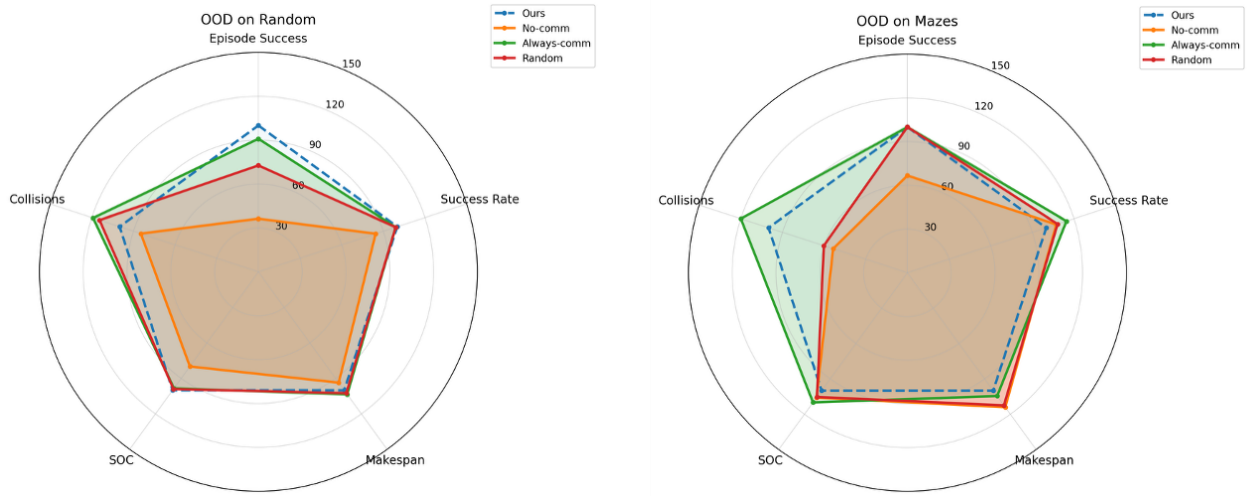
Figure 6: Comparison with IC3Net, TarMAC, and MAPPO across Random, Mazes, and Warehouse. Our method (blue dashed) achieves the largest radar area in all three environments. IC3Net shows particularly poor Makespan performance, while TarMAC falls behind on Collisions.

Figure 6 compares our method against IC3Net [3], TarMAC [4], and MAPPO [9] (equivalent to No-comm). The most striking pattern is the shape of IC3Net's radar polygon: it achieves moderate Success Rate but collapses on Makespan across all three environments. This is consistent with our hypothesis that a gate trained indirectly from sparse task rewards may struggle to distinguish timesteps where communication prevents a slow resolution from timesteps where it does not. This can lead to either over-suppression, where important coordination moments are missed, or over-activation, where conflicting messages cause agents to wait. The net effect is an increase in the time for the last agent to finish.

TarMAC performs closer to our method on Success Rate and Episode Success but shows higher Collision counts, particularly in Warehouse. This suggests that attending to relevant neighbors (TarMAC's contribution) helps global coordination but does not prevent local conflicts, because attending to a neighbor's message does not guarantee that the message is actionable at that particular moment. Our gate adds this second filter: it suppresses the message when the agent is already confident, reducing the noise that inflates collision counts.

Our method achieves the largest radar polygon area in all three environments, showing consistent improvements across all five metrics. Neither IC3Net nor TarMAC achieves this pattern in any single environment.

7.5 Out-of-Distribution Generalization and Scalability



(a) Random (b) Mazes

Figure 7: Out-of-distribution evaluation on held-out Random and Mazes maps. Our method (blue dashed) maintains the largest or near-largest radar area under distribution shift.

Out-of-distribution generalization. Figure 7 evaluates policies on Random and Mazes maps that were not used during training. In the OOD Random setting (left panel), our method, Always-comm, and Random-comm achieve comparable radar areas, all clearly outperforming No-comm on Episode Success. The performance gap between communicating strategies is small because Random maps do not impose the structured interaction patterns that differentiate selective communication from always-on communication. This is consistent with the in-distribution finding of Section 7.2.

The OOD Mazes setting (right panel) reveals a clearer advantage. No-comm degrades substantially on the Collisions axis, as agents trained on simpler maps lack the learned avoidance behavior that narrow passages require. Always-comm and Random-comm achieve moderate collision performance but show inflated SOC, suggesting they adopt overly conservative strategies when faced with unfamiliar corridor geometries. Our method maintains the best Collisions performance among all four strategies while preserving competitive SOC and Success Rate. This generalization advantage arises because the gate's criterion asks whether a message reduces the current decision entropy. This criterion remains meaningful under new map geometries without requiring environment-specific recalibration.

Table 1: Success rate as team size increases. The model is trained with 8 agents; results for 16, 32, and 64 agents are zero-shot transfer evaluations.

Number of Agents	Success Rate
8	0.988
16	0.988
32	0.963
64	0.975

Scalability. Table 1 reports success rate as team size scales from the training size of 8 agents up to 64 agents. Performance remains above 0.96 throughout despite the quadratic growth in potential pairwise interactions. The slight dip at 32 agents (0.963) may reflect increased contention at intermediate density, where the fixed communication radius begins to include more relevant neighbors and the gate must filter a more crowded aggregated message. The partial recovery at 64 agents (0.975) is consistent with a denser neighborhood providing a stronger collective signal that the gate can exploit, even as the raw number of candidates grows.

The combination of top-k neighbor selection and the student gate keeps the per-agent communication load bounded. Each agent attends to at most k neighbors and activates the gate only on a fraction of timesteps, so the total message traffic does not scale proportionally with N. This design choice is what allows performance to remain stable rather than degrading as team size grows.

7.6 Summary

Five analyses yield the following findings. First, uncertainty reduction converges to near-zero during training, but the gate stabilizes at a positive activation rate that scales with environment complexity (0.25–0.35 on Random and 0.5–0.6 on Warehouse). This suggests that communication need is structural rather than a simple proxy for raw uncertainty. Second, always-on communication can increase SOC and Makespan by inducing overly conservative behavior, while selective communication avoids this by gating out messages that do not change the decision. Third, our method achieves a Pareto improvement over Always-comm in the CommCost–Episode Success space on Warehouse maps. Fourth, IC3Net's reward-based gate struggles on Makespan across all environments, while TarMAC's attention-only approach increases Collision counts in dense settings. Combining attention-based selection with uncertainty-based gating addresses both failure modes. Fifth, the gate's criterion generalizes to unseen map geometries without recalibration, yielding the best OOD Collision performance on Mazes while maintaining competitive success and efficiency metrics.

8 Conclusion

We proposed an uncertainty-guided selective communication framework for cooperative multi-agent reinforcement learning under partial observability. The central insight is that communication usefulness should be measured by whether message fusion reduces decision uncertainty, rather than by the absolute level of local uncertainty alone. Building on this idea, our method combines local neighbor selection, attention-based message construction, a teacher-supervised student gate, and a residual distributional critic that makes the value contribution of communication explicit.

Experiments in POGEMA across Random, Mazes, and Warehouse maps support three main findings. First, the gate stabilizes at an activation rate that increases with environment complexity, approximately 0.25–0.35 on Random maps and 0.5–0.6 on Warehouse maps. This suggests that communication need is structural rather than a simple proxy for raw uncertainty. Second, always-on communication can increase SOC and Makespan by inducing overly conservative behavior, while selective communication avoids this by suppressing messages that do not change the decision. Third, the method achieves a Pareto improvement over always-on communication in the CommCost–Episode Success space on Warehouse maps, and its uncertainty-based gate criterion generalizes to unseen map geometries without environment-specific recalibration.

Several limitations point to directions for future work. On the methodological side, the current teacher signal uses immediate uncertainty reduction as a proxy for communication value. Incorporating longer-horizon utility, such as whether a communicated message improves cumulative return over future timesteps, could provide a richer training signal. Separately, the residual distributional critic decomposes value into a local base and a communication-induced residual, but it does not further disentangle environmental stochasticity from uncertainty caused by missing information about other agents. A finer decomposition could improve both interpretability and gate accuracy. On the application side, the current evaluation assumes reliable, zero-latency message delivery. Extending the framework to settings with communication delay, message drop, or limited bandwidth would be an important step toward deployment in real robotic systems with constrained communication channels.

Acknowledgments

I would like to express my sincere gratitude to my advisor and committee chair, Dr. Daniel S. Brown, for his guidance, support, and valuable feedback throughout this master's project. I also thank my committee members, Dr. Tucker Hermans and Dr. Tom Henderson, for their time, feedback, and thoughtful comments. I am grateful to the members of the ARIA Lab for their helpful discussions and support. This work was completed as part of my master's research in Computer Science at the University of Utah.

Code and Project Website The project website and source code are available at https://seongil-heo.com/project/2026ucmarl and https://github.com/SeongilHeo/ucmarl.

References

[1] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[2] 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, 2016.

[3] Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. *arXiv preprint arXiv:1812.09755*, 2018.

[4] 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*, pages 1538–1546. PMLR, 2019.

[5] 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, pages 5142–5149, 2020.

[6] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E. Taylor. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 33(6):750–797, 2019.

[7] Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. Springer, 2016.

[8] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[9] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaming Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems*, volume 35, pages 24611–24624, 2022.

[10] Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pages 449–458. PMLR, 2017.

[11] Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[12] Dapeng Li, Na Lou, Zhiwei Xu, Bin Zhang, and Guoliang Fan. Efficient communication in multi-agent reinforcement learning with implicit consensus generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23240–23248, 2025.

[13] Ling Ding, Tianbai Lyu, Zhiliang Bi, Hao Wang, Shanshan Feng, and Wei Yu. Communication-efficient multi-agent reinforcement learning with spatiotemporal information hub. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 20799–20807, 2026.

[14] Georgy Antonov and Peter Dayan. Exploring uncertainty in distributional reinforcement learning. In *Reinforcement Learning Conference*, 2024.

[15] Somnath Hazra, Pallab Dasgupta, and Soumyajit Dey. Tackling uncertainties in multi-agent reinforcement learning through integration of agent termination dynamics. *arXiv preprint arXiv:2501.12061*, 2025.

[16] Ruoqi Wen, Rongpeng Li, Xing Xu, and Zhifeng Zhao. Multi-agent uncertainty-aware pessimistic model-based reinforcement learning for connected autonomous vehicles. *arXiv preprint arXiv:2503.20462*, 2025.

[17] Alexey Skrynnik, Anton Andreychuk, Anatolii Borzilov, Alexander Chernyavskiy, Konstantin Yakovlev, and Aleksandr Panov. POGEMA: A benchmark platform for cooperative multi-agent pathfinding. *arXiv preprint arXiv:2407.14931*, 2024.