SELF-SUPERVISED INTENT SIGNALS FOR INTRA-STEP MULTI-AGENT COORDINATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Decentralized agents must typically select actions at the same instant from local observations, so no agent can see its teammates' current-step decisions. Centralized training with decentralized execution (CTDE) stabilizes learning, but it does not remove this execution-time ambiguity. Most existing communication protocols transmit encoded trajectories or multi-step forecasts, so each recipient still has to guess what others will do now. We study a two-stage policy that runs inside a single environment timestep while preserving simultaneous execution. Each agent first emits a compact signal trained with a reconstruction objective to retain both its local view and a draft action distribution; agents then form executed actions by conditioning on the collected signals. Task-driven policy gradients are withheld from the message module so that the signals remain semantically stable and can serve as reliable inputs to the joint policy, allowing competing intents to be reconciled before the environment transitions. On fully cooperative suites and team-versus-team tasks against learning opponents, this design consistently improves over recent communication-based MARL methods.

1 INTRODUCTION

Multi-Agent Reinforcement Learning (MARL) has emerged as a powerful method for solving distributed tasks, ranging from traffic control (Jiang et al., 2025) to robotic swarms (Hussein et al., 2022). These problems are typically formalized as Decentralized Partially Observable Markov Decision Processes (Dec-POMDPs) (Bernstein et al., 2002; Oliehoek et al., 2016), where agents must act simultaneously from local observations to optimize a shared objective. Because transitions and rewards depend on the joint action, each agent must commit without seeing what its teammates will do now (Busoniu et al., 2008; Lowe et al., 2017). This execution-time ambiguity is especially costly in bottleneck steps, such as covering the same landmark, firing load on the same food, or occupying the same attacking lane, where two locally reasonable actions become a team failure.

Centralized Training with Decentralized Execution (CTDE) methods such as QMIX (Rashid et al., 2020) and MAPPO (Yu et al., 2022) stabilize learning with privileged centralized training signals, but they do not remove this ambiguity. At execution time, agents still select actions independently, so they remain prone to miscoordination and suboptimal equilibria (Huh & Mohapatra, 2023). Teammate modeling (Wu et al., 2025; Kontogiannis et al., 2025) can reduce the guesswork by inferring others from history, yet it cannot confirm the action that will be taken in the present step.

Communication is the natural remedy, yet most protocols still leave the recipient to guess the current-step action. Perception-sharing methods transmit encoded observations (Sukhbaatar et al., 2016; Lo et al., 2024); intention-sharing methods transmit imagined trajectories or multi-step forecasts (Kim et al., 2020; Guo et al., 2024); sequential-execution methods condition later agents on earlier actions (Ruan et al., 2022; Ding et al., 2024). The first two families lack an explicit draft of the action that will be taken *now*, while the third violates the simultaneous-execution assumption of Dec-POMDPs.

In this paper, we propose Pre-Action Communication through Tentative Decision Coordination (PreACT), a two-stage policy that emits self-supervised intent signals inside a single environment timestep and then forms executed actions from the collected signals, while preserving simultaneous execution. In the first stage, each agent emits a communication signal trained with a reconstruction objective to retain both its local observations and a draft action distribution. In the second stage, agents form executed actions by conditioning on the collected signals, so competing intents can be

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.

reconciled before the environment transitions. The primary risk in this design is semantic drift: if the message is updated by task-driven policy gradients, the meaning of a signal can change from one update to the next and can no longer serve as a reliable input to the joint policy. PreACT therefore withholds those gradients from the message module, so that the signals remain semantically stable.

Our main contributions are summarized as follows: First, we instantiate intra-step coordination as a two-stage policy that broadcasts intent signals and then forms executed actions inside one environment step, without sequential execution. Second, we train those signals with a reconstruction objective over the local observation and the draft action distribution, and isolate the encoder from task-driven policy gradients so the signals stay semantically stable. Third, we show that PreACT improves over recent communication-based MARL methods on cooperative suites and team-versus-team tasks against learning opponents, with interpretable signal semantics and graceful degradation under imperfect channels.

2 Related Work

We first review CTDE methods that learn coordinated policies without an execution-time channel, then organize communication-based MARL by *what* is transmitted: perception, future behavior, or an imposed execution structure.

Centralized Training Decentralized Execution The Centralized Training with Decentralized Execution (CTDE) paradigm is the default training paradigm for cooperative MARL. Value-decomposition methods (VDN (Sunehag et al., 2018), QMIX (Rashid et al., 2020)) factor a joint action-value under a monotonicity constraint; centralized-critic methods (MADDPG (Lowe et al., 2017), MAPPO (Yu et al., 2022), HAPPO (Kuba et al., 2022)) stabilize policy gradients with a privileged critic. In all of these methods, execution remains fully local: each agent maps local perception to the self action with no confirmation of teammates' concurrent choices. Teammate-modeling approaches try to close that gap without a channel, by maintaining public beliefs (BAD (Foerster et al., 2019)), extracting hidden representations (MAIC (Yuan et al., 2022)), actively probing behaviors (AIM (Wu et al., 2025)), estimating shared policies (SMPE (Kontogiannis et al., 2025)), aligning latents (IWoL (Lee et al., 2025)), or mixing task-adaptive opponent intents (Odrowaz-Sypniewski et al., 2026). Inference from history can capture long-term conventions, but it cannot reveal a draft that is formed and possibly revised inside the current step.

Perception Sharing. A first family of communication protocols transmits *what is seen*. Early methods learned implicit continuous vectors end-to-end (CommNet (Sukhbaatar et al., 2016), DIAL (Foerster et al., 2016), ICP (Wang et al., 2025)). Subsequent work optimized the channel itself: gating when to speak (IC3Net (Singh et al., 2019)), attending to whom (TarMAC (Das et al., 2019), T2MAC (Sun et al., 2024), PMAC (Meng & Tan, 2024)), imposing communication-constrained priors (Du-MIE (Yang et al., 2025)), or selecting messages by value of information (VIL2C (Zhang et al., 2026)). To improve semantic quality, recent methods reconstruct a global state from encoded observations via contrastive learning (CACL (Lo et al., 2024), TACTIC (Yu et al., 2025)) or attention aggregation (MASIA (Guan et al., 2022), COCOM (Li et al., 2025), MATCH (Ding et al., 2026)). Sharing perception mitigates partial observability, but the recipient must still infer *how* a teammate will act from the reconstructed view.

Future Behavior Communication. A second family transmits planned future behavior rather than raw percepts. Agents broadcast imagined trajectories (IS (Kim et al., 2020), MAPPO-IS (Guo et al., 2024), CAIC (Li et al., 2026)), high-level goals (SARNet (Rangwala & Williams, 2020), ToM2C (Wang et al., 2022), MARIC (He et al., 2025)), decisive states (UDI (Zhang et al., 2025)), or planned joint actions (MCAS (Asmar & Kochenderfer, 2026)). Reward-independent messaging (Yoshida & Taniguchi, 2025) decouples the channel from the return, which is related to our gradient isolation, but still does not emit a current-step draft action. These protocols are primarily *cross-step* and reactive: the payload is a forecast rolled out from previous experience. If an agent's intent shifts because of a newly observed event in the present step, teammates that rely on that forecast cannot adapt before the transition.

Structural Coordination. A third family reduces simultaneous conflicts by imposing an execution order or a restricted interaction graph. GCS (Ruan et al., 2022) builds graph-based execution, DAG-BN (Chen & Zhang, 2023) models context-aware action dependencies, SeqComm (Ding et al., 2024) conditions later agents on earlier actions, and GRDC (Gao et al., 2026) restricts messages to intra-role graphs. The structure shrinks the joint search space, but it breaks the simultaneous-execution assumption of Dec-POMDPs and adds inference latency growing with team size.

Positioning PreACT. CTDE leaves current-step actions unconfirmed; perception sharing lacks action semantics; intention sharing is cross-step; structural methods sacrifice simultaneous execution. PreACT occupies the remaining intra-step gap. Each agent broadcasts a compact intent signal that a reconstruction objective grounds in the local view and a draft action distribution. Teammates then form executed actions from the collected signals in the same timestep, reconciling competing intents without sequential execution.

3 PRELIMINARIES

In this section, we introduce the definitions of Decentralized Partially Observable Markov Decision Process (Dec-POMDP) and actor-side communication.

3.1 DEC-POMDP

The cooperative multi-agent reinforcement learning (MARL) problem is mainly formalized as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) (Oliehoek et al., 2016), defined by the tuple $\langle \mathcal{N}, \mathcal{S}, \{\mathcal{O}_i\}_{i\in\mathcal{N}}, \{\mathcal{A}_i\}_{i\in\mathcal{N}}, \{\Omega_i\}_{i\in\mathcal{N}}, P, R, \gamma \rangle$. In the tuple, $\mathcal{N} = \{1, \ldots, n\}$ denotes the set of agents and $\mathcal{S}$ represents the state space. At each timestep t, every agent $i \in \mathcal{N}$ receives a local observation $o_t^i \in \mathcal{O}_i$ generated by the observation function $\Omega_i(s_t)$ and subsequently selects an action a_t^i from individual action space $\mathcal{A}_i$. The joint action $\boldsymbol{a}_t = \left(a_t^1, \ldots, a_t^n\right)$ induces a state transition to s_{t+1} by the transition $P(s_{t+1}|s_t, \boldsymbol{a}_t)$, resulting in a shared reward $r_t = R(s_t, \boldsymbol{a}_t)$. The objective is to find a joint policy $\boldsymbol{\pi} = \left\{\pi^i\right\}_{i\in\mathcal{N}}$ that maximizes the expected cumulative discounted reward $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$, where $\gamma \in [0, 1)$ denotes the discount factor and π^i denotes the individual policy.

3.2 ACTOR-SIDE COMMUNICATION

To support intra-step coordination, the standard Dec-POMDP formulation is extended to include a communication mechanism with message space $\mathcal{M} \subseteq \mathbb{R}^{d_M}$, where d_M is the message dimension. At each timestep t, upon acquiring the local observation o_t^i, each agent i generates a communicative message $m_t^i \in \mathcal{M}$ via a message generator $\sigma^i : \mathcal{O}_i \to \mathcal{M}$, which encodes the individual perception into a broadcast signal, resulting in a joint message set $\boldsymbol{m}_t = \left(m_t^1, \ldots, m_t^n\right)$. Consequently, the decision-making process is refined such that the policy $\pi^i : \mathcal{O}_i \times \mathcal{M}^n \to \Delta(\mathcal{A}_i)$ conditions on both the local observation o_t^i and the collective message set $\boldsymbol{m}_t$ to determine the final action a_t^i. The learning objective involves jointly optimizing the message generation functions $\boldsymbol{\sigma} = \left\{\sigma^i\right\}_{i\in\mathcal{N}}$ and action policies $\boldsymbol{\pi}$ to maximize the expected cumulative team reward:

$$\max_{\boldsymbol{\pi},\boldsymbol{\sigma}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \boldsymbol{a}_t)\right], \tag{1}$$

where the expectation accounts for the stochasticity in state transitions, observations, message generation, and action selection induced by the joint strategies.

4 PREACT: SELF-SUPERVISED INTENT SIGNALS

We propose Pre-Action Communication through Tentative Decision Coordination (PreACT), a two-stage policy that emits self-supervised intent signals inside one environment step and then forms executed actions from the collected signals (Figure 1). Both stages complete before the next state transition is queried, so execution remains simultaneous. Stage 1 produces a compact intent signal

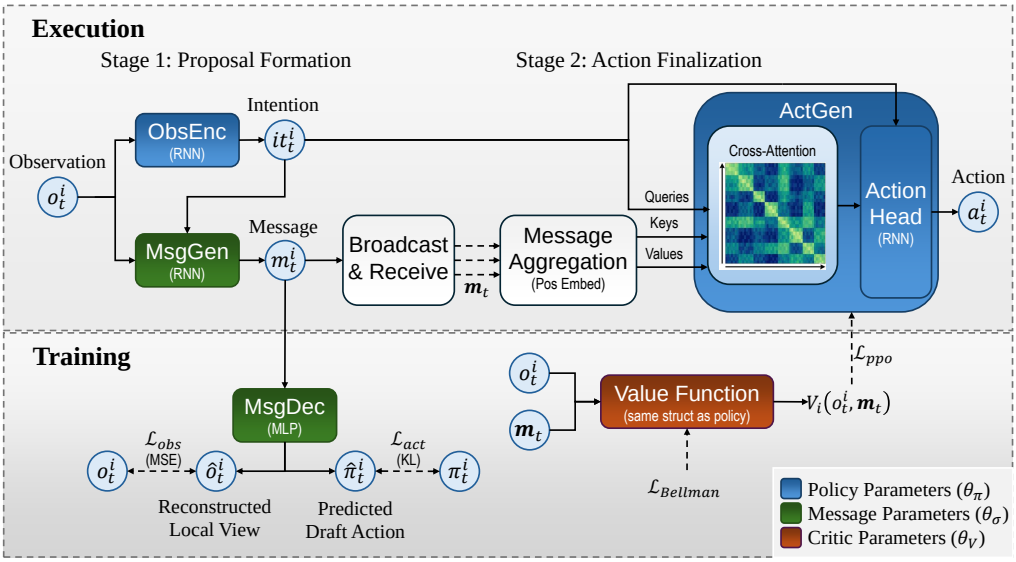


Figure 1: Overview of PreACT. The two-stage policy decomposes into an Observation Encoder (ObsEnc_i) and an Action Generator (ActGen_i). Stage 1: ObsEnc_i maps o_t^i to an intent embedding it_t^i, and the Message Generator (MsgGen_i) emits an intent signal m_t^i. Stage 2: ActGen_i forms the executed action a_t^i from it_t^i and the collected signals $\boldsymbol{m}_t$. During training only, a Message Decoder (MsgDec_i) reconstructs the local view and a draft action distribution, while the critic $V_i(o_t^i, \boldsymbol{m}_t)$ conditions on the same signals. See Algorithm 1 in Appendix A.

from the local view and a local policy prior; Stage 2 conditions the executed action on every agent's signal based the policy prior. The remainder of this section specifies the protocol, the reconstruction objective with gradient isolation, and policy optimization.

4.1 Two-Stage Intra-Step Communication Protocol

We split π^i into an Observation Encoder (ObsEnc_i) and an Action Generator (ActGen_i). This split is what makes intra-step coordination possible: the encoder can form an intention before any peer signal is observed, and the generator can revise the intention after the collected signals arrive. At timestep t, ObsEnc_i maps the local observation to an *intent embedding*

$$it_t^i = \mathrm{ObsEnc}_i\left(o_t^i\right) \in \mathbb{R}^{d_H}. \tag{2}$$

Because ObsEnc_i is the prefix of π^i that processes o_t^i before any collected signals $\boldsymbol{m}_t$ are observed, the intent embedding it_t^i already carries the semantics of a draft action as a hidden state; it is not an explicit sample or distribution over $\mathcal{A}_i$.

The Message Generator (MsgGen_i) then emits an *intent signal* $m_t^i \in \mathbb{R}^{d_M}$ from both the raw observation and this intent embedding:

$$m_t^i = \mathrm{MsgGen}_i\left(o_t^i, it_t^i\right). \tag{3}$$

Including o_t^i keeps the channel anchored in a relatively stable observation even as the evolving policy changes it_t^i given the same local perception. Without the raw observation, a fixed observation would produce drifting signals as the encoder parameters change, and teammates could not treat m_t^i as a comparable input across updates. The resulting signal is a compact summary of what the agent sees and what it is prepared to do.

The Action Generator then forms the executed action from the intent embedding and the collected signals $\boldsymbol{m}_t = \left(m_t^1, \ldots, m_t^n\right)$, allowing the agent to dynamically adjust its execution based on the

early intent signals of teammates:

$$\begin{aligned}\pi^i\left(\cdot\middle|o_t^i, \boldsymbol{m}_t\right) &= \mathrm{ActGen}_i\left(it_t^i, \boldsymbol{m}_t\right) \\ &= \mathrm{ActGen}_i\left(\mathrm{ObsEnc}_i\left(o_t^i\right), \boldsymbol{m}_t\right).\end{aligned} \tag{4}$$

Parameters of ObsEnc_i and ActGen_i are updated jointly as the policy, so task-driven policy gradients do flow into ObsEnc_i. Those gradients are withheld from MsgGen_i, which is trained only by the reconstruction objective below.

4.2 Self-Supervised Intent Signal Reconstruction

An intent signal must retain the sender's local perception and a *draft action distribution*—what the agent will likely do unless intra-step coordination revises it. We pair MsgGen_i with a training-only Message Decoder MsgDec_i that maps m_t^i back to a reconstructed observation $\hat{o}_t^i$ and a predicted action distribution $\hat{\pi}_t^i$:

$$\left(\hat{o}_t^i, \hat{\pi}_t^i\right) = \mathrm{MsgDec}_i\left(m_t^i\right). \tag{5}$$

The reconstruction objective is

$$\mathcal{L}_{AE} = \lambda_{obs}\mathcal{L}_{obs} + \lambda_{act}\mathcal{L}_{act} = \lambda_{obs}\left\|o_t^i - \hat{o}_t^i\right\|^2 + \lambda_{act} D_{KL}\left(\hat{\pi}_t^i \middle\| \mathrm{detach}\left(\pi_t^i\left(\cdot\middle|o_t^i, \boldsymbol{m}_t\right)\right)\right), \tag{6}$$

where $\mathcal{L}_{obs}$ is a mean squared error (Lee, 1998) on the local observation and $\mathcal{L}_{act}$ distills a detached copy of the executed policy into the decoder. We treat $\hat{\pi}_t^i$ as the student and $\pi_t^i(\cdot \mid o_t^i, \boldsymbol{m}_t)$ as the teacher, and use $D_{KL}(\hat{\pi}_t^i \| \pi_t^i)$ rather than the reverse. This direction is mode-seeking: the decoder is penalized for placing mass where the policy does not, so m_t^i is driven to encode a peaked draft (the action the agent is prepared to take) instead of covering the full support of π^i. The reverse KL $D_{KL}(\pi^i \| \hat{\pi}^i)$ is mean-seeking and would force the signal to explain every low-probability mode, which injects ambiguity into intra-step coordination; Appendix C.4 shows a small but consistent advantage for the default direction.

Withholding task-driven policy gradients from MsgGen_i is as important as the reconstruction terms themselves. If MsgGen_i were updated by the task reward, the meaning of m_t^i would drift with every policy step and could not serve as a semantically stable input to ActGen_i; empirically this collapse looks like CommNet (Section 5.3). The teacher is also deliberately asymmetric: $\pi_t^i(\cdot \mid o_t^i, \boldsymbol{m}_t)$ conditions on all signals, yet MsgDec_i must match it from m_t^i alone. Distilling a post-coordination policy through a local decoder therefore stores a marginalized draft in the signal, i.e., what the agent will likely do unless coordinated otherwise, rather than an action that already depends on teammates.

4.3 Signal Aggregation and Policy Optimization

To form the executed action, ActGen_i aggregates collected signals with multi-head cross-attention (Vaswani et al., 2017). The intent embedding it_t^i is the query; $\boldsymbol{m}_t$ supplies keys and values. Positional embeddings (PosEmb) preserve sender identity, which is needed for role allocation:

$$c_t^i = \mathrm{Attention}\left(it_t^i, \boldsymbol{m}_t + \mathrm{PosEmb}, \boldsymbol{m}_t + \mathrm{PosEmb}\right). \tag{7}$$

The design is permutation-invariant and applies to variable team sizes. The context c_t^i is concatenated with it_t^i and passed through an RNN (Cho et al., 2014; Sherstinsky, 2020) that tracks coordination history, so an agent can remember how a previous intra-step revision was resolved.

The policy is optimized with independent PPO (De Witt et al., 2020): there is no parameter sharing and the critic does not observe the global state. We do, however, condition the individual value function on the collected signals, $V_i\left(o_t^i, \boldsymbol{m}_t\right)$, so that advantage estimates reflect teammates' draft intents rather than local observations alone. This keeps training decentralized while still letting the critic evaluate the situation after intra-step coordination.

5 Experiments

In this section, we evaluate PreACT on cooperative and team-versus-team MARL benchmarks. We structure our experimental analysis to answer the following research questions: empirical performance (Q1), design of the intent-signal objective (Q2), channel robustness and bandwidth (Q3), and whether the signals are semantically stable enough to reconcile competing intents (Q4).

Q1: Does PreACT improve intra-step coordination and empirical performance?

Q2: Do reconstruction and gradient isolation make the signals useful?

Q3: How does PreACT behave under noisy, delayed, or bandwidth-limited channels?

Q4: Do the learned signals encode draft actions and support intra-step coordination?

5.1 Experimental Setup

Environments. We evaluate three suites that stress different forms of intra-step coordination; details and reward definitions are in Appendix B.

Multi-Agent Particle Environment (MPE) (Mordatch & Abbeel, 2018) is a continuous-space benchmark under partial observability. We use four team-reward variants: `Cooperative Navigation` (N agents cover N landmarks), where identical agents must assign themselves to distinct targets; `Rounded Speaker-Listener` (a cyclic information chain), where each agent knows only a neighbor's goal; `Physical Deception` (N agents deceive an adversary); and `Predator-Prey` (N agents encircle M faster prey). In the last two, adversaries are co-trained with independent PPO (De Witt et al., 2020) and no communication, so the cooperative team must coordinate against a non-stationary opponent.

Level-Based Foraging (LBF) (Papoudakis et al., 2021) is a grid-world in which food is collected only if adjacent agents execute `load` in the same timestep and the sum of their levels meets the food's level. Partial observability (limited sight) and a shared team reward make this a strict test of simultaneous execution: arriving one step apart yields nothing. We use two `coop` maps in which every food needs at least two agents. We evaluate on the `4s-12x12-5p-3f-coop` (sparse rewards) and `5s-15x15-7p-7f-coop` (scaled coverage) scenarios.

Google Research Football (GRF) (Kurach et al., 2020) is a physics-based football simulator with a 115-dimensional observation and 19 discrete actions (moves, pass, shoot, sprint). Coordination here is both micro (timing a pass with an off-ball run) and macro (occupying lanes before the defense recovers). We test on `3_vs_1_with_keeper` (precise passing) and `counterattack_hard` (rapid numerical exploitation) scenarios.

Baselines. We benchmark PreACT against diverse methods representing different MARL paradigms to ensure a comprehensive evaluation: (1) **MAPPO** (Yu et al., 2022): The standard CTDE baseline. (2) **MAPPO-IS** (Kim et al., 2020; Guo et al., 2024): An intention-sharing method where agents communicate imagined future trajectories. (3) **SMPE** (Kontogiannis et al., 2025): A non-communication agent-modeling approach that explores the actions and global information of others. (4) **CACL** (Lo et al., 2024): An observation-oriented method where agents communicate encoded observations to reconstruct the global state. (5) **SeqComm** (Ding et al., 2024): An action-coordination approach utilizing sequential execution to condition policies on preceding actions. (6) **UDI-Comm** (Zhang et al., 2025): A communication method that modifies the state-based intent target representation into messages. We also compare with RGMComm (Chen et al., 2024), HAPPO (Kuba et al., 2022), IPPO (De Witt et al., 2020), SARNet (Rangwala & Williams, 2020), IC3NET (Singh et al., 2019) and CommNet (Sukhbaatar et al., 2016), which underperform the representative baselines in their respective categories. See Appendix C for additional experimental results. All experimental results are averaged over four distinct random seeds, with the shaded regions in the figures representing the standard error (Everitt & Skrondal, 2010).

5.2 Empirical Performance (Q1)

We evaluate PreACT's coordination capability across cooperative and competitive scenarios. The learning curves are illustrated in Figures 2 and 3.

In the MPE scenarios, PreACT excels in spatial coordination, adversarial adaptability, and scalability. In Cooperative Navigation, it scales robustly across different team sizes ($N = 5, 7, 15$): intra-step coordination reconciles spatial conflicts before the transition, preventing the "lazy-agent" and collision behaviors often observed in centralized methods (MAPPO, CACL, SMPE). In Speaker-Listener, encoding a draft action in the intent signal yields faster convergence than communication baselines. In competitive settings, PreACT coordinates spatial dispersion to act as decoys in Physical

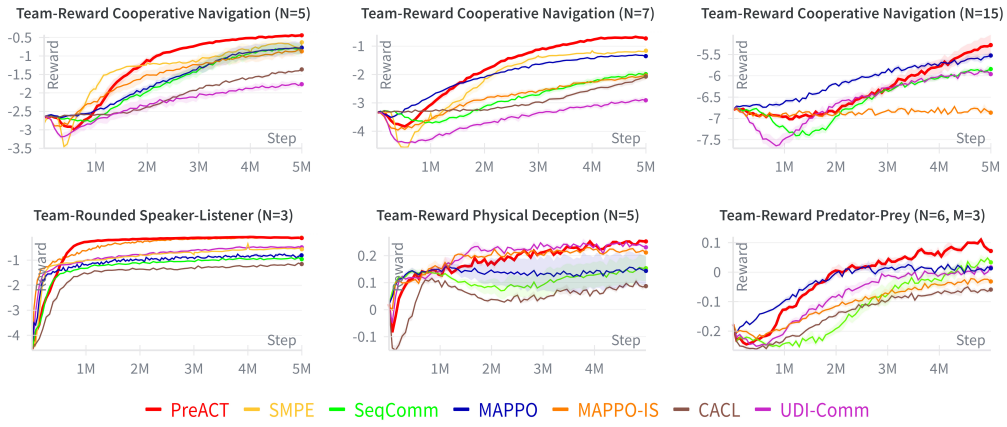


Figure 2: Performance comparison in MPE environment.

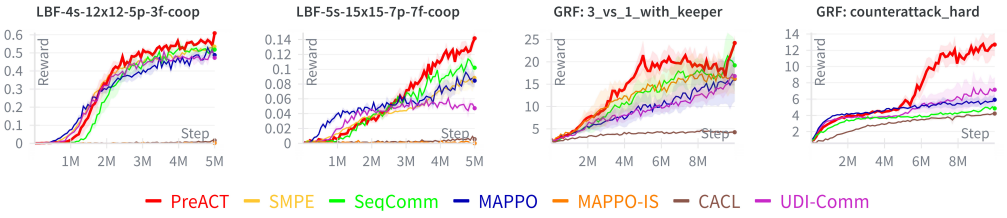


Figure 3: Performance comparison in LBF and GRF environment.

Deception and forms encirclement traps to capture faster prey in Predator-Prey, whereas baselines often collapse into suboptimal clustering or endless chasing.

In the LBF grid-world tasks, the strict simultaneous execution severely penalizes asynchronous policies. In sparse-reward `4s-12x12-5p-3f-coop`, PreACT's intent signals carry exploration drafts, overcoming credit assignment failures of MAPPO-IS and UDI-Comm to enable simultaneous `load` actions. In the scaled `5s-15x15-7p-7f-coop` scenario, reconciling competing target drafts avoids redundant clustering, achieving multi-target coverage where centralized learning fails.

In the high-dimensional GRF environment, PreACT demonstrates superior micro- and macro-coordination. In `3_vs_1_with_keeper`, PreACT and SeqComm achieve comparable leading performance with lower variance than centralized baselines. Crucially, in the highly dynamic `counterattack_hard` scenario, while baselines (MAPPO, CACL) produce overcrowded, easily intercepted attacks, PreACT's intra-step coordination distributes attackers into distinct lanes to stretch the defense and exploit numerical superiority.

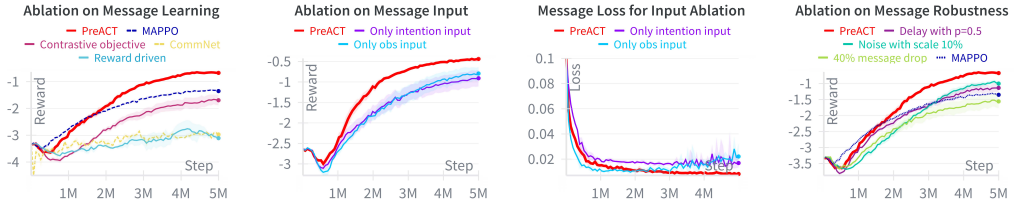


Figure 4: Ablation on message learning objective, message input, and communication channel.

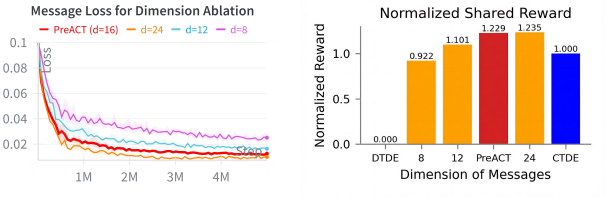


Figure 5: Message loss and reward of different dimensions.

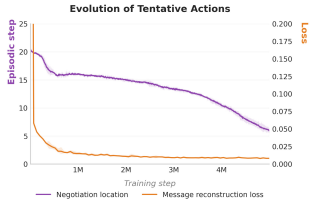


Figure 6: Evolution of negotiation.

5.3 Intent-Signal Design (Q2)

A core principle of PreACT is that effective coordination requires intent signals to retain the agent's local view and a draft action. We validate this through two key ablations of MsgGen_i: its learning objective and its input formulation.

Self-Supervised Objective vs. Alternatives. We compare our self-supervised objective ($\mathcal{L}_{AE}$) against a reward-driven baseline that updates MsgGen_i solely via policy gradients and a contrastive baseline that maximizes mutual information to form a global representation (Parulekar et al., 2023), as shown in Figure 4. PreACT consistently outperforms both. Reward-driven signal semantics drift severely, collapsing to CommNet's performance. Contrastive learning captures global information but lacks the structure of a draft action, performing slightly worse than MAPPO. By reconstructing both the local view and the draft action, with task-driven gradients withheld from MsgGen_i, PreACT keeps the signals semantically stable for intra-step coordination rather than merely inferring the global state.

Input Design for Stability. We ablate the dual-input design (o_t^i, it_t^i) against single-source variants. As shown in Figure 4, both degrade performance. The reconstruction loss spikes in later training stages, which confirms that intent embeddings derived from evolving policies are non-stationary. Feeding it_t^i together with o_t^i lets the input co-evolve with the target policy, maintaining a consistent input-to-target mapping and stabilizing early representation learning.

5.4 Channel Robustness and Efficiency (Q3)

Real-world applications often operate under imperfect communication and limited bandwidth. To assess PreACT's practical viability, we investigate its resilience to channel disruptions and its efficiency in signal dimension.

Channel Robustness. As shown in Figure 4, under Gaussian noise and 50% stochastic delays, PreACT still outperforms MAPPO, indicating robust embeddings and temporal continuity. Against a severe 40% random drop rate, PreACT's cross-attention reweights the available signals, and agents fall back to local priors. Instead of catastrophic failure, PreACT gracefully degrades to non-communicative MAPPO performance.

Message Dimension. In Figure 5, performance is normalized such that IPPO (DTDE) and MAPPO (CTDE) correspond to 0 and 1, respectively. Ablating the signal dimension $d_M \in \{8, 12, 16, 24\}$ reveals that higher dimensions accelerate reconstruction and steadily improve final performance. Notably, $d_M = 12$ is already sufficient to surpass the CTDE baseline. Given that CTDE relies on transmitting full global observations, our default $d_M = 16$ compresses the local view and draft action into a compact latent vector.

5.5 Interpretability and Coordination Analysis (Q4)

To understand the mechanism underlying the improvements, we analyze how PreACT reduces execution-time ambiguity and locks in coordinated conventions.

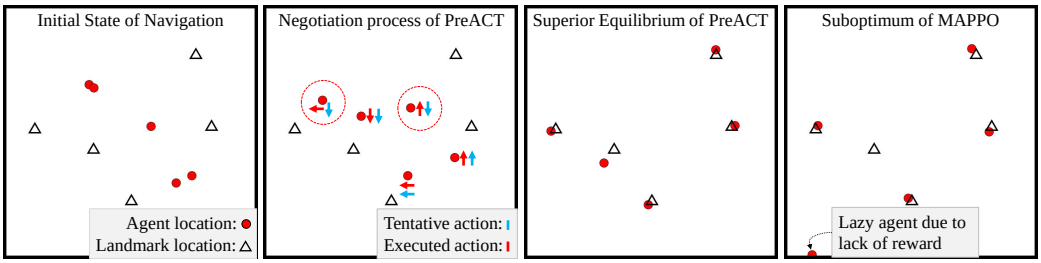


Figure 7: Example of the negotiation process.

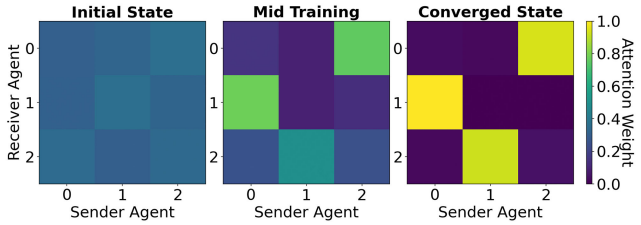


Figure 8: Evolution of communication attention heatmaps.

Coordination via Semantically Stable Signals. In Dec-POMDPs, local observations lack information about teammates' current-step actions, preventing standard algorithms from forming stable policies. PreACT addresses this by sharing intent signals that remain semantically stable. As shown in Figure 7, overlapping spatial drafts are revised inside the step, so agents adjust execution before the environment transitions. This preemptively resolves conflicts, locking in better conventions and avoiding MAPPO's typical collision or "lazy-agent" behaviors.

When Coordination Happens. In Figure 5 (right), the decay of reconstruction loss indicates that agents quickly learn to broadcast readable drafts. A persistent non-zero floor reflects the divergence between local priors and executed actions after seeing $\boldsymbol{m}_t$, which is exactly the intra-step revision. Building on these signals, PreACT uses reward-driven optimization of the policy to discover conventions. Because executed actions are conditioned on intent signals rather than local observations alone, PreACT avoids the non-stationarity that plagues standard MARL. This process is corroborated by the curve of when coordination is most active: as conventions stabilize, revision shifts to the start of the episode, so later steps require little adjustment.

Attention Mechanism. Figure 8 visualizes the evolution of cross-attention weights in ActGen on Rounded Speaker-Listener. Over training, the mechanism assigns high weight to the relevant speaker. Agents therefore filter the channel and condition executed actions on the partner's draft rather than on an average of all signals.

6 Conclusion

We introduced PreACT, a two-stage policy that emits self-supervised intent signals inside one environment step and then forms executed actions from the collected signals. A reconstruction objective retains the local view and a draft action distribution, while policy gradients are withheld from the message module so the signals stay semantically stable. Across cooperative suites and team-versus-team tasks, this intra-step design improves over recent communication-based MARL methods.

Limitation and Future Direction. The present implementation broadcasts to the full team. Because mature policies concentrate coordination early in the episode, a natural extension is to learn *when* and with *whom* to emit a signal. Additionally, a promising direction would be to extend the framework with asynchronous communication protocols and multi-turn negotiations.

AI USE STATEMENT

We used generative AI to edit the manuscript for readability and to modify small code modules and assist with debugging. We have reviewed all AI-assisted text and code, and take responsibility for the final content of this work.

ETHICS STATEMENT

This work studies cooperative multi-agent reinforcement learning in public simulated environments (Multi-Agent Particle Environment, Level-Based Foraging, and Google Research Football). It does not involve human subjects, personally identifiable information, or sensitive datasets. The proposed communication protocol is evaluated only in simulation. We do not identify immediate harmful applications beyond the generic dual-use concerns of coordination algorithms. The authors have read and adhere to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We provide the materials needed to reproduce our results. Algorithm 1 specifies the training algorithm. Network architectures, training objectives, and PreACT hyperparameters are reported in Appendix A (Table 1); baseline hyperparameters are in Table 2. Environment configurations and reward definitions appear in Appendix B. Additional learning curves, ablations, and representation analyses appear in Appendix C.

REFERENCES

Dylan M Asmar and Mykel J Kochenderfer. Efficient multiagent planning via shared action suggestions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29314–29322, 2026.

Richard Bellman. Dynamic programming. *science*, 153(3731):34–37, 1966.

Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4): 819–840, 2002.

Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

Dingyang Chen and Qi Zhang. Context-aware bayesian network actor-critic methods for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 5327–5350. PMLR, 2023.

Jingdi Chen, Tian Lan, and Carlee Joe-Wong. Rgmcomm: Return gap minimization via discrete communications in multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17327–17336, 2024.

Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.

Filippos Christianos, Lukas Schäfer, and Stefano Albrecht. Shared experience actor-critic for multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:10707–10717, 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. PMLR, 2019.

Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

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

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, pp. 20799–20807, 2026.

Brian S Everitt and Anders Skrondal. *The Cambridge dictionary of statistics*, volume 4. Cambridge university press Cambridge, UK, 2010.

Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

Jakob Foerster, Francis Song, Edward Hughes, Neil Burch, Iain Dunning, Shimon Whiteson, Matthew Botvinick, and Michael Bowling. Bayesian action decoder for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 1942–1951. PMLR, 2019.

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. *Advances in Neural Information Processing Systems*, 35:1020–1033, 2022.

Yicheng Guo, Jiaqi Liu, Rongjie Yu, Peng Hang, and Jian Sun. Mappo-pis: A multi-agent proximal policy optimization method with prior intent sharing for cavs' cooperative decision-making. In *European Conference on Computer Vision*, pp. 244–263. Springer, 2024.

Xin He, Hongwei Ge, Zhangang Hao, and Yaqing Hou. Maric: an efficient multi-agent real-time intention-based communication model for team cooperation. *Neural Computing and Applications*, 37(5):3925–3942, 2025.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Dom Huh and Prasant Mohapatra. Multi-agent reinforcement learning: A comprehensive survey. *arXiv preprint arXiv:2312.10256*, 2023.

Aya Hussein, Eleni Petraki, Sondoss Elsawah, and Hussein A Abbass. Autonomous swarm shepherding using curriculum-based reinforcement learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 633–641, 2022.

Jiaxuan Jiang, Ling Pan, Lin Zhou, Longbo Huang, and Zhixuan Fang. Tackling sparsity in designated driver dispatch with multi-agent reinforcement learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1060–1069, 2025.

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Andreas Kontogiannis, Konstantinos Papathanasiou, Yi Shen, Giorgos Stamou, Michael M. Zavlanos, and George Vouros. Enhancing cooperative multi-agent reinforcement learning with state modelling and adversarial exploration. In *Forty-second International Conference on Machine Learning*, 2025.

JG Kuba, R Chen, M Wen, Y Wen, F Sun, J Wang, and Y Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *ICLR 2022-10th International Conference on Learning Representations*, pp. 1046. The International Conference on Learning Representations (ICLR), 2022.

Karol Kurach, Anton Raichuk, Piotr Stańczyk, Michał Zając, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, et al. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 4501–4510, 2020.

Dongsu Lee, Daehee Lee, Yaru Niu, Honguk Woo, Amy Zhang, and Ding Zhao. Unifying agent interaction and world information for multi-agent coordination. In *NeurIPS 2025 Workshop: Second Workshop on Aligning Reinforcement Learning Experimentalists and Theorists*, 2025.

Te-Won Lee. Independent component analysis. In *Independent component analysis: Theory and applications*, pp. 27–66. Springer, 1998.

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, pp. 23240–23248, 2025.

Pei Li, Yongkang Zhang, Zhonglin Lv, Jinmin Zhu, and Jiangjin Yin. CAIC: Congestion-aware intent communication for multi-agent reinforcement learning. In Emilija Perković and Daniel Malinsky (eds.), *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, volume 337 of *Proceedings of Machine Learning Research*, pp. 3705–3718. PMLR, 17–21 Aug 2026. URL `https://proceedings.mlr.press/v337/li26h.html`.

Yat Long Lo, Biswa Sengupta, Jakob Nicolaus Foerster, and Michael Noukhovitch. Learning multi-agent communication with contrastive learning. In *International conference on learning representations (ICLR)*, 2024.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Xiangrui Meng and Ying Tan. Pmac: Personalized multi-agent communication. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17505–17513, 2024.

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, 2018.

Mateusz Odrowaz-Sypniewski, Jasmine Bayrooti, Ajay Shankar, and Amanda Prorok. Generalized intention modeling in multi-agent reinforcement learning. *arXiv preprint arXiv:2605.31318*, 2026.

Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS)*, 2021.

Advait Parulekar, Liam Collins, Karthikeyan Shanmugam, Aryan Mokhtari, and Sanjay Shakkottai. Infonce loss provably learns cluster-preserving representations. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 1914–1961. PMLR, 2023.

Murtaza Rangwala and Ryan Williams. Learning multi-agent communication through structured attentive reasoning. *Advances in Neural Information Processing Systems*, 33:10088–10098, 2020.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

Jingqing Ruan, Yali Du, Xuantang Xiong, Dengpeng Xing, Xiyun Li, Linghui Meng, Haifeng Zhang, Jun Wang, and Bo Xu. Gcs: Graph-based coordination strategy for multi-agent reinforcement learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, pp. 1128–1136, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450392136.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. In *International Conference on Learning Representations (ICLR)*, 2019.

Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29, 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.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2085–2087, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Han Wang, Binbin Chen, Tieying Zhang, and Baoxiang Wang. Learning to communicate through implicit communication channels. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Yuanfei Wang, Jing Xu, Yizhou Wang, et al. Tom2c: Target-oriented multi-agent communication and cooperation with theory of mind. In *International Conference on Learning Representations (ICLR)*, 2022.

Hao Wu, Shoucheng Song, Chang Yao, Sheng Han, Huaiyu Wan, Youfang Lin, and Kai Lv. Think how your teammates think: Active inference can benefit decentralized execution. *arXiv preprint arXiv:2511.18761*, 2025.

Guang Yang, Tianpei Yang, Jingwen Qiao, Yanqing Wu, Jing Huo, Xingguo Chen, and Yang Gao. Multi-agent reinforcement learning with communication-constrained priors. In D. Belgrave, C. Zhang, H. Lin, L. Montoya, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 154040–154060. Curran Associates, Inc., 2025.

Naoto Yoshida and Tadahiro Taniguchi. Reward-independent messaging for decentralized multi-agent reinforcement learning. In *International Conference on Neural Information Processing*, pp. 367–382. Springer, 2025.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 2022.

Peihong Yu, Manav Mishra, Syed Zaidi, and Pratap Tokekar. Tactic: Task-agnostic contrastive pre-training for inter-agent communication, 2025.

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.

Qian Zhang, Zhuo Sun, Yao Zhang, Zhiwen Yu, Bin Guo, and Jun Zhang. Vil2c: value-of-information aware low-latency communication for multi-agent reinforcement learning. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence and Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence and Sixteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'26/IAAI'26/EAAI'26. AAAI Press, 2026. ISBN 978-1-57735-906-7. doi: 10.1609/aaai.v40i35.40234. URL https://doi.org/10.1609/aaai.v40i35.40234.

Zeren Zhang, Zhiwei Xu, Guangchong Zhou, Dapeng Li, Bin Zhang, and Guoliang Fan. Unveiling decision intention for cooperative multi-agent reinforcement learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pp. 2345–2354, 2025.

A Implementation Details

In this section, we provide a comprehensive description of the PreACT algorithm implementation to ensure reproducibility. We detail the hardware infrastructure, the software specifications, the specific neural network architectures, the complete training pipeline, and the exact hyperparameter configurations used for all reported baselines.

A.1 Hardware Infrastructure

To ensure reproducibility and provide context for the computational requirements of the PreACT framework, we detail the hardware infrastructure used for all experiments:

- **CPU:** The system is powered by dual AMD EPYC 9654 96-Core Processors, yielding a total of 192 physical cores (384 logical threads).
- **GPU:** Deep learning optimization and neural network inference were accelerated using a multi-GPU setup comprising four NVIDIA L20 GPUs. Each GPU provides 48 GB of VRAM (192 GB total).
- **System Memory:** The machine is equipped with approximately 768 GB of system memory (RAM).

A.2 Software Specification

The software environment was containerized using Conda to maintain dependency consistency. The primary deep learning framework is PyTorch 2.3.1 running on Python 3.11.5. The multi-agent simulation environments rely on OpenAI Gym 0.26.2. Key dependencies include NumPy 1.23.2 and SciPy 1.13.0 for numerical computation, mpi4py 4.1.1 for distributed process management, and CUDA 12.1 with cuDNN 8.9.2 for GPU acceleration.

A.3 Network Architecture

The PreACT framework employs a modular architectural design that separates the decision-making process into distinct functional units. We categorize the network parameters into three primary groups: the policy parameters θ_π^i, which govern the Observation Encoder (ObsEnc_i) and the Action Generator (ActGen_i); the communication parameters θ_σ^i, which manage the Message Generator

(MsgGen_i) and the Message Decoder (MsgDec_i); and the critic parameters θ_V^i. During the execution phase, the agent utilizes ObsEnc_i, ActGen_i, and MsgGen_i, while MsgDec_i and θ_V^i are employed exclusively during training. We detail the specific layer configurations for each module in the following.

A.3.1 Observation Encoder

The Observation Encoder (ObsEnc_i) serves as the initial processing unit, responsible for transforming the raw local observation $o_t^i \in \mathcal{O}_i$ into an intent embedding $it_t^i \in \mathbb{R}^{d_H}$, where d_H is the hidden dimension for all layers. We adopt a recurrent policy structure to handle partial observability through a sequential processing pipeline. First, the observation is passed through a Multi-Layer Perceptron (MLP) comprising two fully connected layers to extract immediate spatial features. These features are subsequently fed into a Gated Recurrent Unit (GRU) layer (Cho et al., 2014) to integrate temporal information. The GRU output constitutes the agent's intent embedding it_t^i, encapsulating both the historical environmental perception and the preliminary policy calculation for decision-making.

A.3.2 Message Generator

The Message Generator (MsgGen_i) constructs the intent signal m_t^i that is broadcast to other agents. Instead of using raw observations directly, it takes the concatenation of the local observations and the generated intent embedding it_t^i as input. This design ensures that the message encodes both stable spatial perception and the evolving policy intent. The concatenated feature is processed through an MLP block and a GRU layer to maintain temporal consistency in the communication channel. The final output is linearly projected to the compact message dimension d_M.

A.3.3 Action Generator

The Action Generator (ActGen_i) is responsible for synthesizing the local intent embedding with the collected intent signals from teammates to form the executed action. This module is composed of a message perception block and an action refinement block.

To effectively process the incoming joint message set $\boldsymbol{m}_t$, we implement a *Message Cross-Attention* mechanism. The agent's local intent embedding it_t^i serves as the Query (Q), while the received intent signals function as Keys (K) and Values (V). Since standard cross-attention mechanisms are natively permutation-invariant, we introduce explicit positional embeddings to preserve the sender's identity. The attention mechanism is mathematically defined as follows:

- *QKV Projections with Positional Embeddings:* The query is projected into the attention dimension via a linear layer and Layer Normalization. For the messages, each message $m_t^j \in \boldsymbol{m}_t$ is first projected via a linear layer, Layer Normalization, and GELU activation to form a base representation M_j. To distinguish signals from different agents, we add a learnable positional embedding scaled by a factor α_{pos}:

$$\begin{aligned} Q_i &= \mathrm{LayerNorm}(W_Q \cdot it_t^i), \\ K_j, V_j &= M_j + \alpha_{pos} \cdot E_{pos}(j), \end{aligned} \tag{8}$$

  where $E_{pos}(j) \in \mathbb{R}^{d_{attn}}$ is the explicit positional embedding for the j-th agent slot, initialized with a small standard deviation. The scaling factor α_{pos} (initialized to 0.001) ensures that the positional signals do not overwhelm the core message content during early training.
- *Attention and Residual Connection:* The context vector is computed using standard multi-head dot-product attention over Q_i, K_j, V_j. To ensure robust gradient flow for the raw communication signals and preserve information when attention weights are sparse, we additionally apply a lightweight residual connection. The flattened concatenation of all incoming messages is linearly projected and added to the attention output, weighted by a hyperparameter λ_{res}.

The final context vector c_t^i generated by this augmented attention mechanism is concatenated with the original intent embedding it_t^i. This combined feature vector is then processed by a subsequent GRU layer. This secondary refinement step allows the agent to update its internal state based on the collected signals before passing the features to the final action head (Linear-Softmax) to produce the executed-action distribution $\pi(\cdot|o_t^i, \boldsymbol{m}_t)$.

A.3.4 MESSAGE DECODER

The Message Decoder (MsgDec_i) is an auxiliary network utilized to enforce semantic fidelity in generated messages. It features a shared trunk followed by two parallel heads: an observation reconstruction head and an action prediction head. The network processes the message m_t^i to reconstruct the original observation $\hat{o}_t^i$ and a draft action distribution $\hat{\pi}_t^i$. We employ MLP with Layer Normalization and GELU activations (Hendrycks & Gimpel, 2016) throughout the decoder to ensure stable optimization of the self-supervised reconstruction objective.

A.3.5 VALUE FUNCTION

The Value Function network (V_i) provides for policy optimization by estimating the expected return $V_i(o_t^i, \boldsymbol{m}_t)$. To mitigate the non-stationarity introduced by the communication channel, the critic is designed to explicitly condition its evaluation on the collected intent signals of the team. The architecture mirrors that of the policy network to ensure that the critic has sufficient representational capacity. First, the local observation is encoded into a latent critic state via an MLP followed by a GRU layer, similar to the observation encoder. We subsequently employ the exact same *Message Cross-Attention* module (with Positional Embeddings) to aggregate coordination information. In this context, the critic's latent state serves as the Query (Q), attending to the joint message set $\boldsymbol{m}_t$ (Keys and Values) to extract relevant coordination signals. The resulting context vector is concatenated with the latent state and processed through a fusion MLP block. Finally, a secondary GRU layer integrates the temporal dynamics of the value estimate before a linear output head produces the scalar value prediction.

A.4 TRAINING OBJECTIVES

We optimize network parameters using Independent Proximal Policy Optimization (IPPO) (Schulman et al., 2017; De Witt et al., 2020). The total objective consists of three distinct components: the policy loss, the value function loss, and the communication reconstruction loss.

Generalized Advantage Estimation: To reduce the variance of the policy gradient estimates, we compute the advantage function $\hat{A}_t$ using Generalized Advantage Estimation (GAE) (Schulman et al., 2015). The advantage is defined as the exponentially weighted sum of temporal difference (TD) errors:

$$\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \, \delta_{t+l}, \quad \text{where} \quad \delta_t = r_t + \gamma V_{\theta_V}\left(o_{t+1}^i, \boldsymbol{m}_{t+1}\right) - V_{\theta_V}\left(o_t^i, \boldsymbol{m}_t\right). \tag{9}$$

Here, γ is the discount factor, λ is the GAE smoothing parameter, and δ_t represents the TD error of the value function V_{θ_V}.

Policy Loss ($\mathcal{L}_{ppo}$): We utilize the clipped surrogate objective to enforce stable policy updates. Let $r_t(\theta) = \frac{\pi_\theta(a_t|o_t^i, \boldsymbol{m}_t)}{\pi_{\theta_{old}}(a_t|o_t^i, \boldsymbol{m}_t)}$ denote the probability ratio between the current and old policies. The policy loss minimizes the following objective:

$$\mathcal{L}_{ppo} = -\mathbb{E}\left[\min\left(r_t(\theta)\hat{A}_t, \mathrm{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right] - \beta S\left[\pi_{\theta_\pi}\right], \tag{10}$$

where ϵ is the clipping hyperparameter (set to 0.1) that bounds the update step size, $S\left[\pi_{\theta_\pi}\right]$ is the entropy bonus to encourage exploration, and β is the entropy coefficient.

Value Loss ($\mathcal{L}_{Bellman}$): The critic minimizes the Mean Squared Error (MSE) between the estimated value and the target returns R_t (Bellman, 1966; Schulman et al., 2015). To strictly limit the change in value estimates between updates, we employ a clipped value loss:

$$\mathcal{L}_{Bellman} = \mathbb{E}\left[\max\left(\left(V_{\theta_V} - R_t\right)^2, \left(V_{\theta_V}^{clip} - R_t\right)^2\right)\right], \tag{11}$$

where V_{θ_V} is the current value prediction. The clipped value $V_{\theta_V}^{clip}$ is constrained around the old value estimate $V_{\theta_{V,old}}$ within the range ϵ:

$$V_{\theta_V}^{clip} = V_{\theta_{V,old}} + \mathrm{clip}\left(V_{\theta_V} - V_{\theta_{V,old}}, -\epsilon, \epsilon\right). \tag{12}$$

This mechanism ensures that the value function does not diverge due to excessively large gradient steps in a single batch.

Communication Reconstruction Loss ($\mathcal{L}_{AE}$): To encourage that the broadcast intent signals m_t^i faithfully encapsulate both the agent's local view and a draft action, we employ a self-supervised auto-encoding objective. The Message Decoder (MsgDec_i) reconstructs the local observation $\hat{o}_t^i$ and the draft action distribution $\hat{\pi}_t^i$ from the message. The loss is computed as:

$$\mathcal{L}_{AE} = \lambda_{obs} \underbrace{\left|o_t^i - \hat{o}_t^i\right|^2}_{\mathcal{L}_{obs}\text{: Observation MSE}} + \lambda_{act} \underbrace{D_{KL}\left(\hat{\pi}_t^i \middle| \mathrm{detach}\left(\pi_t^i(\cdot|o_t^i, \boldsymbol{m}_t)\right)\right)}_{\mathcal{L}_{act}\text{: Action Fidelity}}. \tag{13}$$

Here, D_{KL} denotes the Kullback-Leibler divergence. Minimizing this objective aligns the decoder's predicted draft action distribution $\hat{\pi}_t^i$ with the executed-action distribution π_t^i, ensuring that the intent signal is a dense representation of the agent's draft action.

A.5 Training Pipeline

The training follows a Decentralized Training with Decentralized Execution (DTDE) paradigm augmented by the two-stage intra-step protocol. We emphasize that there is no parameter sharing among agents; each agent i maintains and updates its own set of parameters $\{\theta_\pi^i, \theta_V^i, \theta_\sigma^i\}$ using only its local observations o_t^i and the shared intent signals. Algorithm 1 outlines the complete information flow.

A.6 Hyperparameter Settings

We utilize Adam optimizer (Kingma & Ba, 2014), hidden dimension at 64, and communication dimension at 16 for all methods to ensure fair comparison. We present the detailed hyperparameter configurations as shown in Table 1.

Table 1: Hyperparameter settings for **PreACT**.

Hyperparameter	**Value**
Optimization	
Policy Learning Rate	3×10^{-4}
Critic Learning Rate (MPE-LBF/GRF)	$1 \times 10^{-2} - 1 \times 10^{-3}$
Message Learning Rate	1×10^{-3}
PPO Configuration	
Number of Rollouts (MPE-LBF/GRF)	128 - 64
Number of Mini-batches	1
PPO Epochs	15
Clip Parameter (ϵ)	0.1
Entropy Coefficient (β) (MPE-LBF-GRF)	0.02 - 0.01 - 0.003
GAE Parameter (λ)	0.95
Discount Factor (γ)	0.99
Gain	0.01
Attention & Architecture	
Attention Heads (MPE-LBF/GRF)	2 - 1
Residual Weight (λ_{res}) (MPE-LBF/GRF)	0.3 - 0.5
Dropout Rate	0.05
Hidden Dimension (d_H)	64
Message Dimension (d_M)	16
Observation Reconstruction Weight (λ_{obs})	1
Action Estimation Weight (λ_{act})	1

B Detailed Experimental Setup

This section outlines the detailed specifications for the multi-agent environments utilized in our evaluation, spanning three diverse frameworks: the Multi-Agent Particle Environment (MPE) (Mordatch

Algorithm 1 PreACT Training Procedure

Require: Environment $\mathcal{E}$, agents $\mathcal{N} = \{1, \ldots, N\}$, timesteps T, episode Length L, hyperparameters $\alpha_\pi, \alpha_V, \alpha_\sigma$

Initialize independent policy parameters θ_π^i, critic parameters θ_V^i, message parameters θ_σ^i for all $i \in \mathcal{N}$
Initialize independent Replay Buffers $\mathcal{B}_i$ for all $i \in \mathcal{N}$
while total steps $t < T$ **do**
  Randomly reset environment $\mathcal{E}$, obtain initial observations $\{o_t^i\}_{i \in \mathcal{N}}$
  for episode step $s = 0$ to L **do**
    // Stage 1: emit intent signal
    for agent $i \in \mathcal{N}$ **do**
      Compute intent embedding $it_t^i = \mathrm{ObsEnc}_i(o_t^i | \theta_\pi^i)$
      Generate intent signal $m_t^i = \mathrm{MsgGen}_i(o_t^i, it_t^i | \theta_\sigma^i)$
      Broadcast m_t^i
    end for
    Collect joint intent signals $\boldsymbol{m}_t = \left\{ m_t^j \right\}_{j \in \mathcal{N}}$
    // Stage 2: execute action
    for agent $i \in \mathcal{N}$ **do**
      Sample executed action $a_t^i \sim \mathrm{ActGen}_i(it_t^i, \boldsymbol{m}_t | \theta_\pi^i)$
    end for
    Execute joint action $\boldsymbol{a}_t$, receive team reward r_t and next obs $\{o_{t+1}^i\}_{i \in \mathcal{N}}$
    Store transition $(o_t^i, it_t^i, \boldsymbol{m}_t, a_t^i, r_t, o_{t+1}^i)$ in $\mathcal{B}_i$ for all $i \in \mathcal{N}$
    // Optimization Phase
    if update condition met **then**
      for agent $i \in \mathcal{N}$ **do**
        // Update Agent i independently using $\mathcal{B}_i$
        Compute value estimates $v_k^i = V_i(o_k^i, \boldsymbol{m}_k)$ using batch from $\mathcal{B}_i$
        Compute GAE advantages $\hat{A}^i$ using v_k^i
        Update θ_V^i minimizing $\mathcal{L}_{Bellman}$
        Update θ_π^i minimizing $\mathcal{L}_{ppo}$
        Recompute $m_t^i = \mathrm{MsgGen}_i(o_t^i, it_t^i | \theta_\sigma^i)$, and predict $(\hat{o}_t^i, \hat{\pi}_t^i) = \mathrm{MsgDec}_i(m_t^i | \theta_\sigma^i)$
        Update θ_σ^i minimizing $\mathcal{L}_{AE}$
      end for
    end if
    $t \leftarrow t + 1$
  end for
end while

& Abbeel, 2018), Level-Based Foraging (LBF) (Christianos et al., 2020), and Google Research Football (GRF) (Kurach et al., 2020). Across these benchmarks, we enforce strict cooperative settings. For the MPE tasks, we modify the standard structures to a *Team-Reward* configuration to exacerbate the difficulty of credit assignment and policy coordination. The LBF and GRF environments inherently demand high-level spatio-temporal synchronization and complex tactical execution, respectively.

B.1 Environment Implementation Details

Although the agents in our MPE experiments operate within a discrete action space, the underlying environment simulation is governed by a continuous physics engine. At each timestep, the discrete actions selected by the agents (typically corresponding to movement in cardinal directions) are mapped to physical force vectors. These forces are applied to the agents—modeled as particles with mass—to compute instantaneous acceleration. The state of the environment, specifically the position p and velocity v of each agent, is updated via numerical integration:

$$v_{t+1} = v_t \cdot (1 - \delta_{damp}) + \frac{F_t}{m} \Delta t, \quad p_{t+1} = p_t + v_{t+1} \Delta t, \tag{14}$$

where F_t is the applied force derived from the action, m is the agent's mass, and δ_{damp} is a damping factor. This hybrid structure presents a fundamental challenge: agents must learn to control continuous physical dynamics (momentum and inertia) through coarse, discrete decisions.

B.2 TEAM-REWARD COOPERATIVE NAVIGATION

The Cooperative Navigation task models a classic coverage problem where N agents must cooperate to occupy N distinct landmarks distributed within a bounded 2D space.

Observation: The observation vector o_i for agent i is a concatenation of:

1. Agent i's self-state (absolute position $p_i \in \mathbb{R}^2$).
2. Relative positions of landmarks $\{l_k - p_i\}$ for all k, masked to zero if $\|l_k - p_i\|_2 > 1.0$.
3. Relative positions of other agents $\{p_j - p_i\}$ for all $j \neq i$, masked to zero if $\|p_j - p_i\|_2 > 1.0$.

Setup and Reward. We evaluate scenarios with N agents in a $[-2, 2]^2$ space. The team receives a shared reward based on the proximity of the agents to the landmarks, calculated as the negative sum of the logarithm of the minimum distance from each landmark to the nearest agent:

$$r_t = -\sum_{j=1}^{N} \log\left(1 + \min_i |p_{j,\text{landmark}} - p_{i,\text{agent}}|\right). \tag{15}$$

Using a logarithmic reward function incentivizes agents to achieve high-precision positioning rather than merely coarse proximity.

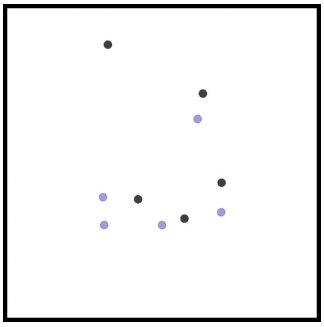


Figure 9: An example of Team-Reward Cooperative Navigation. Here every blue particle denotes an agent, every black particle denotes a landmark.

Key Challenge: Coordination Ambiguity. The primary challenge in this environment is resolving the *coordination ambiguity* inherent in the shared reward signal. Since agents are identical and the reward is global, they must implicitly assign which agent covers which landmark without an explicit assignment protocol. A failure in coordination typically leads to redundant coverage (two agents moving to the same landmark) or uncovered targets, requiring agents to infer the intent of their teammates solely from their observations.

B.3 TEAM-ROUNDED SPEAKER-LISTENER

We introduce a rounded variation of the classic Speaker-Listener task. In a system of N agents, the communication structure forms a closed loop: Agent i observes the target landmark for Agent $(i + 1) \bmod N$, but is blind to its own target.

Observation: Each agent i is assigned a listener $i + 1$ and possesses knowledge of $i + 1$'s target landmark. The observation o_i includes:

1. Agent i's self-state (absolute position p_i).
2. The relative position of the assigned listener $p_j - p_i$ (globally visible).

3. The relative position of the listener's target landmark $T_j - p_i$ (globally visible).
4. Relative positions of all landmarks and other agents within the sensing radius of 1.0.

Setup and Reward. Agents must navigate to their specific assigned landmarks. The system is governed by a sparse shared reward:

$$r_t = -\sum_{i=1}^{N} |p_{i,\text{agent}} - p_{i,\text{target}}|^2. \tag{16}$$

Because agents cannot see their own goals, their movement is entirely dependent on the instructions received from the previous agent in the cycle.

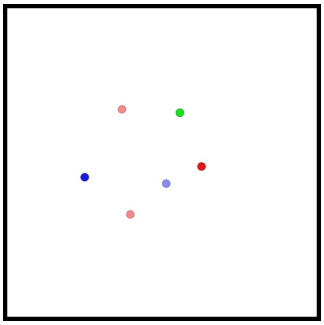


Figure 10: An example of Team-Rounded Speaker-Listener. Here dark particle denotes a landmark, every light particle denotes an agent with color the same as its target.

Key Challenge: Information Asymmetry and Trust. This environment tests the agents' ability to bridge extreme *information asymmetry*. The challenge is two-fold: first, the "speaker" must effectively encode spatial information into abstract messages; second, the "listener" must learn to trust and act upon these messages despite having no direct visual confirmation of the goal. The cyclic dependency means that any break in the communication chain results in a total system failure, demanding robust and simultaneous multilateral communication.

B.4 Team-Reward Physical Deception with Learnable Adversaries

The Physical Deception environment presents a competitive-cooperative challenge involving N cooperative agents and 1 adversarial agent. There are N landmarks, one of which is the true target known only to the cooperative team.

Observation (Cooperative Agents): The observation vector includes the self-state (absolute positions), the relative position of the true target landmark (globally visible), and the relative positions of other entities within the sensing radius.

Observation (Adversary): The adversary observes its self-state (absolute positions) and nearby entities within the sensing radius. Crucially, the vector slot corresponding to the target landmark's relative position is masked with a zero vector, forcing the adversary to infer the target from the movements of the cooperative agents.

Setup and Reward. The cooperative agents aim to reach the target while concealing its identity from the adversary (trained via IPPO (De Witt et al., 2020)). The reward function balances these conflicting objectives:

$$r_t = -\min_{i \in \text{coop}} |p_i - p_{\text{target}}| + \alpha \cdot |p_{\text{adv}} - p_{\text{target}}|, \tag{17}$$

where the coefficient $\alpha = 0.5$ in this work, while the adversary is rewarded by

$$r_{adv,t} = -|p_{\text{adv}} - p_{\text{target}}|. \tag{18}$$

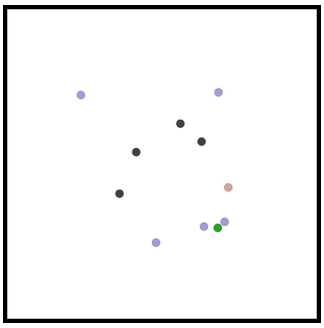


Figure 11: An example of Team-Reward Physical Deception. Here black particle denotes a landmark, every blue particle denotes a good agent, and the red particle denotes the adversary. The target is marked in green color.

Key Challenge: Strategic Deception. The core challenge is *strategic deception*. Simply rushing to the target is a suboptimal strategy as it reveals the target's location to the adversary. Therefore, agents must learn sophisticated role allocation: while one agent approaches the true target, others must convincingly behave as decoys, approaching false landmarks to split the adversary's attention. This requires dynamic role allocation and the ability to model the adversary's belief state.

B.5 Team-Reward Predator-Prey with Learnable Adversaries

The Predator-Prey environment involves a team of N slower predators (cooperative agents) pursuing M faster preys (adversaries). The environment is cluttered with large obstacles to facilitate complex maneuvers.

Observation: Both predators and preys observe their own self-state (absolute positions), the relative positions of landmarks within the sensing radius, and the relative positions of other agents (both teammates and opponents) within the sensing radius. There is no explicit target designation; agents must distinguish friends based on policy logic or implicit order features embedded in the simulator.

Setup and Reward. The predators share a team reward based on the distance to the preys and successful captures (collisions decided by a distance threshold of radius):

$$r_t = -\sum_{m=1}^{M} \min_{i \in \text{pred}} |p_{m,\text{prey}} - p_{i,\text{pred}}| + \beta \cdot N_{\text{collisions}}, \tag{19}$$

where the coefficient $\beta = 0.2$. The preys, possessing higher speed ($v_{max} = 1.2$ vs 1.0) and acceleration ($acc = 3.5$ vs 3.0), are rewarded for avoiding capture via IPPO training (De Witt et al., 2020):

$$r^m_{adv,t} = \min_{i \in \text{pred}} |p_{m,\text{prey}} - p_{i,\text{pred}}|. \tag{20}$$

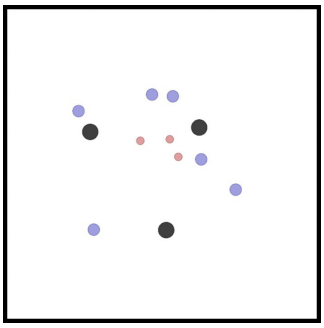


Figure 12: An example of Team-Reward Predator-Prey. Here black particle denotes a landmark, every blue particle denotes a good agent, and every red particle denotes an adversary.

Key Challenge: Coordination against Agility. The defining challenge here is overcoming the *agility gap*. Since individual pursuit is futile against a faster opponent, predators must exhibit high-level spatial coordination to execute “trapping" or “cornering" strategies. They must anticipate the prey’s escape routes and position themselves to cut off these paths, effectively converting the open pursuit into a siege. This tests real-time strategic planning and the ability to maintain formation under high-speed dynamics.

B.6 LEVEL-BASED FORAGING (LBF)

Level-Based Foraging (LBF) (Christianos et al., 2020) is a mixed-cooperative grid-world environment focusing on ad-hoc coordination, where agents are placed in a grid and must collect randomly scattered food items.

Observation: Agents observe the grid within a defined sight radius (field of view). The observation includes the egocentric coordinates and skill levels of the observing agent and other agents within sight, as well as the positions and difficulty levels of nearby food items.

Setup and Reward. Each agent i is assigned a skill level l_i, and each food item f_j has a difficulty level l_{f_j}. We utilize two specific `coop` configurations, meaning food levels are deliberately set such that at least two agents are required to lift any item:

1. **`Foraging-4s-12x12-5p-3f-coop-v3`:** A smaller map (12×12) with limited visibility (4×4 sight) containing 5 agents and 3 food items.
2. **`Foraging-5s-15x15-7p-7f-coop-v3`:** A larger map (15×15) with 7 agents and 7 food items.

Upon successfully collecting a food item, participating agents share a reward normalized by the food’s level.

Key Challenge: Spatio-Temporal Synchronization. The fundamental difficulty in LBF is the strict requirement for both spatial alignment and temporal synchronization. A food item is only collected if the sum of the levels of all adjacent agents simultaneously executing the `load` action exceeds the item’s level ($\sum_{i \in \text{loaders}} l_i \geq l_{f_j}$). Agents must coordinate to arrive at the same location and fire the `load` command within the exact same timestep. Furthermore, the `5p-3f` scenario creates a sparse reward setting demanding cooperative exploration, while the `7p-7f` scenario requires efficient task allocation to maximize coverage and prevent redundant clustering around a single target.

B.7 GOOGLE RESEARCH FOOTBALL (GRF)

Google Research Football (GRF) (Kurach et al., 2020) is a physics-based environment simulating association football. It presents a highly challenging continuous control task characterized by high-dimensional state spaces and complex, fast-paced multi-agent interactions.

Observation: Agents receive a 115-dimensional vector observation containing the egocentric coordinates of the ball, the positions and momentum vectors of all active players, and the overall game status.

Setup and Reward. The action space consists of 19 discrete actions, which include movement directions (e.g., top, bottom, left, right) and specific football commands (e.g., pass, shoot, sprint, slide). We evaluate performance on two official Academy scenarios designed to isolate and test specific coordination skills:

1. **`Academy_3_vs_1_with_keeper`:** 3 offensive agents start at the edge of the box and attempt to score against 1 defender and 1 goalkeeper.
2. **`Academy_counterattack_hard`:** A highly dynamic 4 vs 2 scenario (4 attackers vs 1 defender and 1 goalkeeper) simulating a fast break, where agents spawn in a counter-attack formation.

Agents are rewarded for advancing pass and for scoring a goal and face minor shaping penalties or terminal conditions if the ball is lost or the episode times out.

Key Challenge: Micro-Level Precision and Macro-Level Tactics. GRF requires agents to bridge the gap between high-dimensional continuous physics and discrete tactical decisions. In the `3_vs_1` scenario, the primary challenge is micro-level precision: aligning the exact timing of a pass with the receiver's off-ball movement to break the defensive line. Conversely, the `counterattack_hard` scenario introduces intense macro-level tactical demands; agents must make rapid, intra-step decisions to instantly distribute themselves into distinct offensive lanes, maximizing spatial stretching and exploiting their numerical advantage before the defense can organize.

C Full Experimental Results

In this section, we present the complete performance benchmarks across all the evaluated scenarios and provide detailed ablation studies regarding the internal mechanisms of PreACT. We compare our framework against a comprehensive suite of baselines representing various multi-agent coordination paradigms.

C.1 Baseline Descriptions

To ensure a fair and extensive evaluation, we benchmark PreACT against the following methods. We strictly standardize the architectural configurations across all baselines to guarantee a fair comparison; specifically, we fix the communication dimension at $d_M = 16$ for communication-based approaches and unify the backbone structures by utilizing GRU recurrent layers with a hidden dimension of 64:

- **MAPPO** (Yu et al., 2022): A standard Centralized Training with Decentralized Execution (CTDE) method that adapts PPO with a global value function using the concatenation of all observations to estimate the global state.
- **MAPPO-IS** (Kim et al., 2020; Guo et al., 2024): An intention-sharing approach where agents communicate predicted future trajectories to inform teammates.
- **CACL** (Lo et al., 2024): A perception-sharing method utilizing contrastive learning to reconstruct global state information from broadcast observations.
- **SeqComm** (Ding et al., 2024): A coordination framework that enforces sequential execution orders to condition policies on preceding actions.
- **UDI-Comm** (Zhang et al., 2025): An information-theoretic method that compresses value-based intentions into communication messages.
- **SMPE** (Kontogiannis et al., 2025): A teammate modeling approach that infers global states and explores strategies of other agents.
- **RGMComm** (Chen et al., 2024): A label-based communication method focusing on message efficiency, which is trained to approach centralized training behavior.
- **HAPPO** (Kuba et al., 2022): A trust-region method that sequentially updates agents to ensure monotonic improvements in performance.
- **IPPO** (De Witt et al., 2020): A fully independent implementation of PPO without centralized critics or communication.
- **SARNet** (Rangwala & Williams, 2020): A memory-based approach leveraging attention mechanisms to reason about agent interactions over time.
- **IC3NET** (Singh et al., 2019): A gated communication protocol that learns when to communicate to minimize signaling overhead.
- **CommNet** (Sukhbaatar et al., 2016): An early continuous communication method that averages hidden states across the team.

C.2 Baseline Hyperparameter Settings

To facilitate reproducibility and ensure a rigorous comparison, we present the specific hyperparameter configurations for all baseline algorithms in Table 2. We align general experimental settings such as the network hidden dimension with the PreACT configuration to maintain consistency. The table

below details the learning rates and algorithm-specific parameters used in our experiments, with any unspecified parameters defaulting to the standard settings described in Appendix A.6.

Table 2: Hyperparameter settings for baseline algorithms.

Baseline	**Policy LR**	**Critic LR** (MPE-LBF/GRF)	**Message LR**	**Other Parameters**
MAPPO	3×10^{-4}	$1 \times 10^{-2} - 1 \times 10^{-3}$	-	-
MAPPO-IS	3×10^{-4}	3×10^{-3}	-	bptt_len $= 5$, attn_dim $= 64$, attn_head $= 2$
CACL	3×10^{-4}	$3 \times 10^{-3} - 1 \times 10^{-3}$	-	valuenorm $= True$, window $= 5$, $\eta = 0.1$, $\kappa = 0.5$
SeqComm	3×10^{-4}	$1 \times 10^{-2} - 1 \times 10^{-3}$	-	share_obs_dim $= 16$
UDI-Comm	3×10^{-4}	$1 \times 10^{-2} - 1 \times 10^{-3}$	1×10^{-3}	$k = 1$, ema $= 0.99$, loss $= cosine$
SMPE	5×10^{-4}	5×10^{-4}	-	ED_batch $= 8$, ED_int $= 2000$, ED_dim $= 32$, $L_{rec} = 1$
RGMComm	1×10^{-4}	1×10^{-3}	1×10^{-3}	label_shape $= 16$, phase1 = MADDPG
HAPPO	3×10^{-4}	1×10^{-3}	-	-
IPPO	1×10^{-3}	1×10^{-3}	-	-
SARNet	3×10^{-3}	3×10^{-3}	-	key/value/query $= 16$, traj_upd $= 10$
IC3Net	3×10^{-3}	3×10^{-3}	-	$N_{total} = 10$, $P_{arr} \in [0.05, 0.2]$
CommNet	3×10^{-3}	3×10^{-3}	-	-

C.3 Performance on All Benchmarks

In this section, we report the comprehensive learning curves for all baselines. The results demonstrate that PreACT consistently resolves coordination conflicts more effectively than the baselines, particularly in tasks requiring precise role allocation and spatial synchronization.

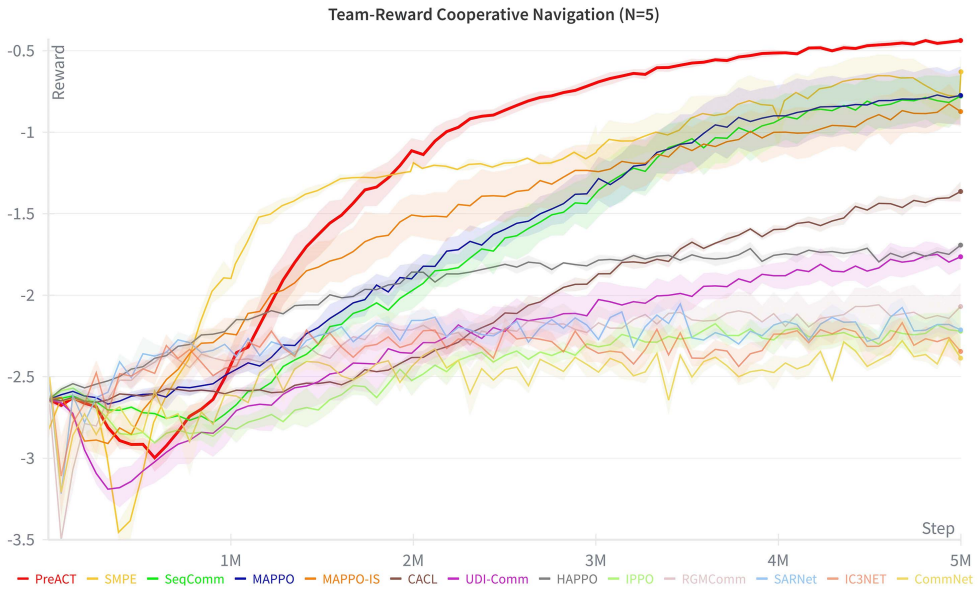


Figure 13: Full Results of Team-Reward Cooperative Navigation ($N = 5$).

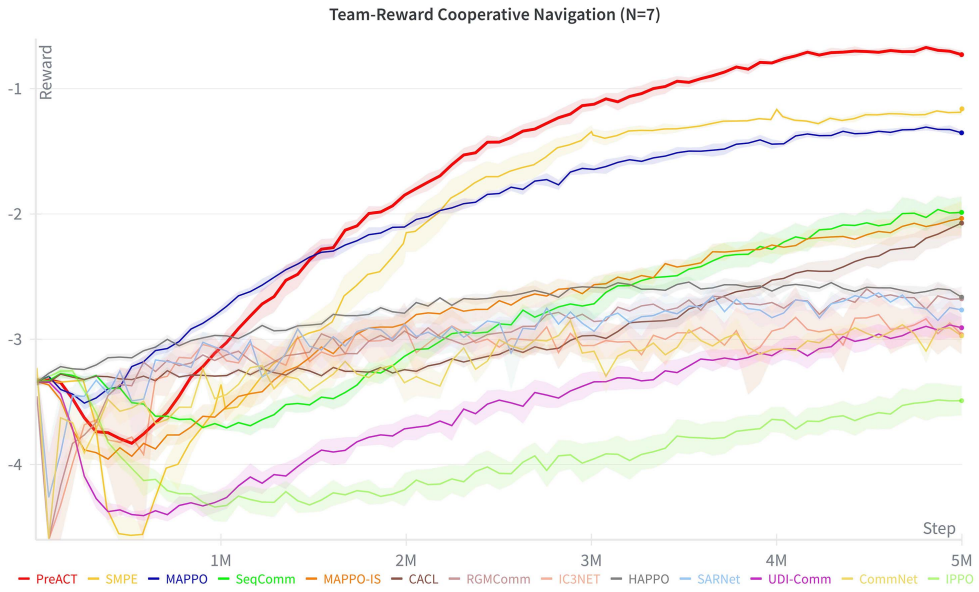


Figure 14: Full Results of Team-Reward Cooperative Navigation ($N = 7$).

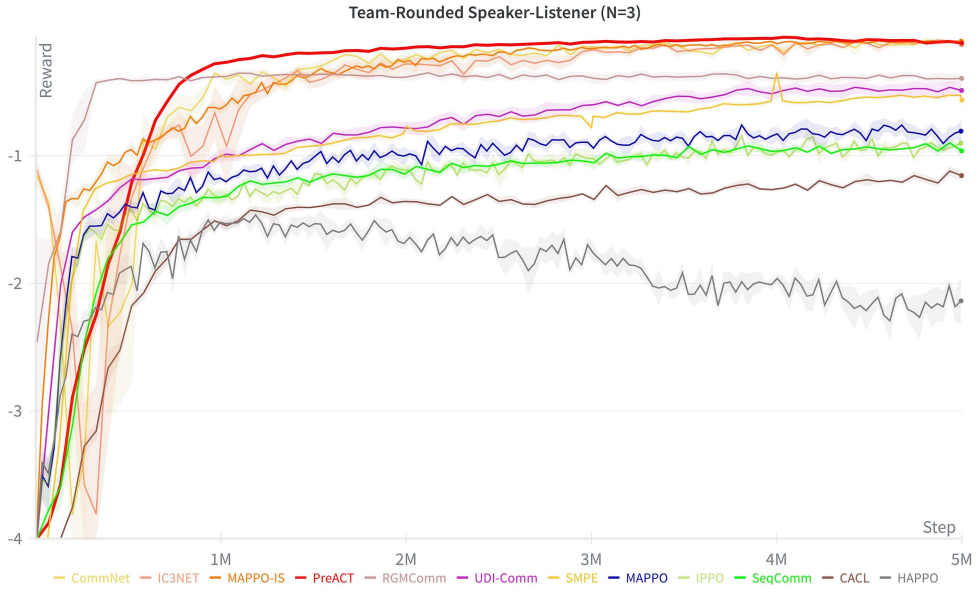


Figure 15: Full Results of Team-Rounded Speaker-Listener ($N = 3$).

In the **Team-Reward Cooperative Navigation** tasks (Figures 13 and 14), the primary challenge lies in the "lazy agent" problem and collision avoidance under a shared team reward. PreACT achieves the highest returns by leveraging intra-step coordination to resolve target conflicts before execution. Among the baselines, SMPE performs robustly, suggesting that explicit teammate modeling can partially substitute for direct communication by allowing agents to infer partners' intents. Standard CTDE methods like MAPPO and intention-sharing approaches like MAPPO-IS perform reliably, but lack the immediate feedback loop to correct redundant movements within the execution timestep. Conversely, observation-sharing methods such as CACL and independent learners like IPPO struggle more significantly; the former burdens agents with inferring policy from raw perception, while

the latter faces inherent difficulties in disentangling individual credit from the global team reward, especially as the agent density increases from 5 to 7.

In the **Team-Rounded Speaker-Listener** task (Figure 15), the environment imposes a high dependency on information exchange. Consequently, PreACT performs on a par with other explicit communication baselines such as SARNet, IC3NET, and CommNet. This parity confirms that the core difficulty here is establishing a stable communication channel rather than complex strategy coordination; once the target information is conveyed, the task is effectively solved. PreACT's advantage lies in its stability and convergence speed, whereas non-communicative baselines (MAPPO, IPPO, and HAPPO) fail to overcome the information asymmetry, as they lack the structural capacity to pass the required goal coordinates.

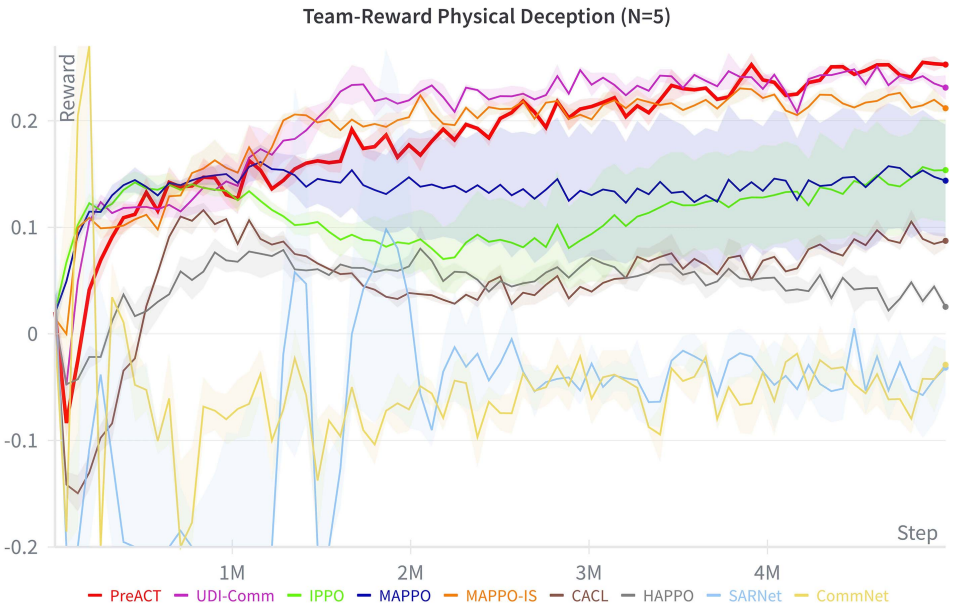


Figure 16: Full Results of Team-Reward Physical Deception ($N = 5$).

The **Team-Reward Physical Deception** task (Figure 16) requires dynamic role assignment to mislead a learning adversary. PreACT, along with MAPPO-IS and UDI-Comm, demonstrates superior performance, indicating that communicating intent is crucial to distinguish between "capturing" and "decoy" roles. While CACL, MAPPO and IPPO can learn basic target-covering strategies, they lack the explicit coordination signals necessary to consistently manipulate the adversary's belief state. SARNet and CommNet lag in this domain, likely because their memory-based reasoning is less effective at adapting to the non-stationary strategies of a co-evolving opponent compared to the direct intent sharing of PreACT.

The **Team-Reward Predator-Prey** task (Figure 17) presents the most significant coordination challenge, requiring agents to form complex encirclements to trap faster prey. PreACT outperforms all competitors by a notable margin, as the intent-signal mechanism allows predators to reconcile competing "cutting off" routes rather than simply chasing. Most baselines, including MAPPO, UDI-Comm, MAPPO-IS, SARNet, and IPPO, tend to converge towards greedy pursuit behaviors; while locally optimal, this strategy fails to capture superior prey. Other baselines like HAPPO and CommNet also struggle, suggesting that merely sharing information is insufficient without the specific ability to form coordinated spatial formations within the same timestep, a capability unique to the PreACT framework.

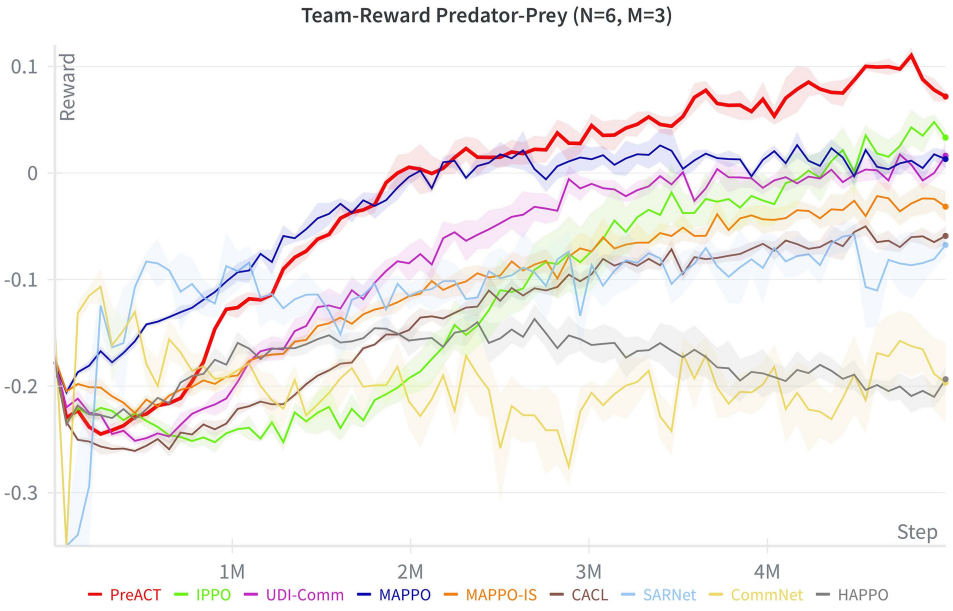


Figure 17: Full Results of Team-Reward Predator-Prey ($N = 6, M = 3$).

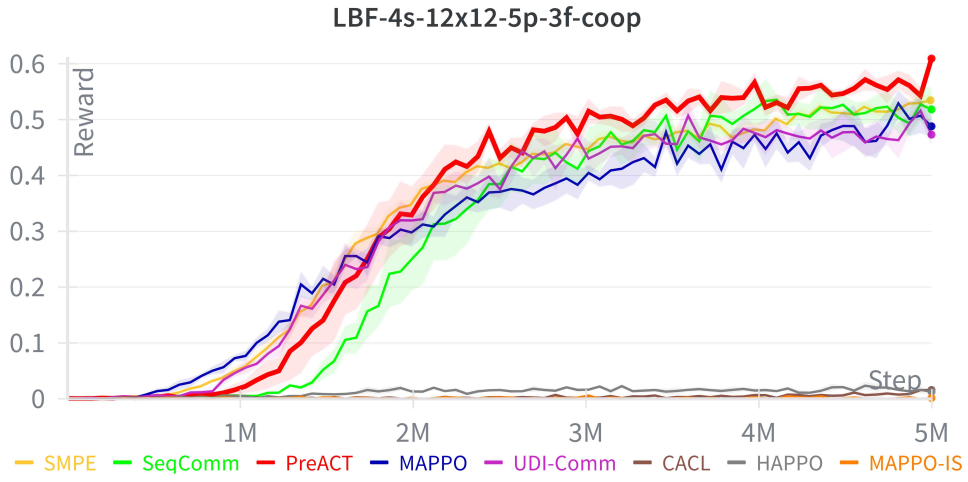


Figure 18: Full Results of `Foraging-4s-12x12-5p-3f-coop-v3`.

As shown in Figure 18, in the `Foraging-4s-12x12-5p-3f-coop-v3` scenario, the primary challenge lies in the sparsity of feedback, which necessitates coordinated exploration to discover and retrieve limited food items. PreACT demonstrates superior performance by leveraging intent signals to convey exploration drafts, effectively mitigating the uncertainty inherent in sparse-reward settings. While SeqComm, MAPPO, UDI-Comm, and SMPE achieve competitive results, methods like CACL, HAPPO, and MAPPO-IS struggle to learn effectively. This failure is likely attributable to their inability to explicitly capture the contribution of cooperative exploration to the team reward, making it difficult to take `load` actions simultaneously around the same food.

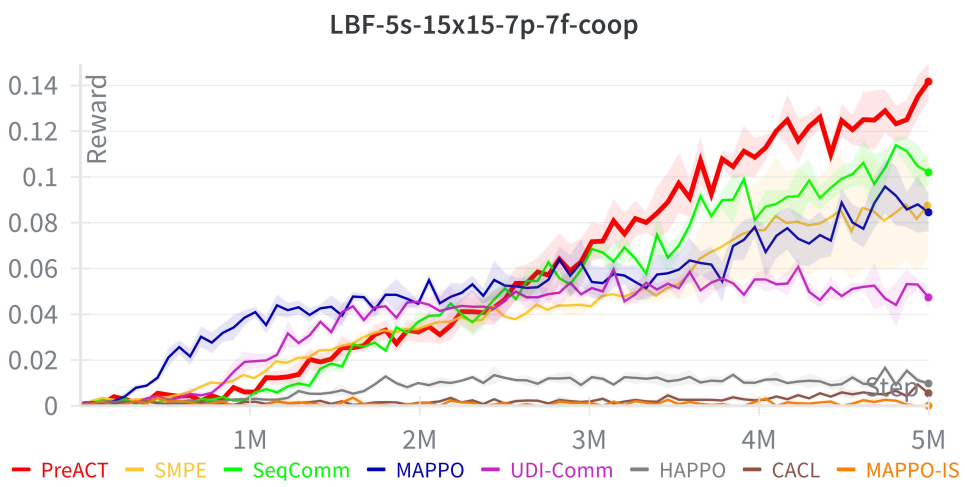


Figure 19: Full Results of `Foraging-5s-15x15-7p-7f-coop-v3`.

In the larger `Foraging-5s-15x15-7p-7f-coop-v3` scenario (Figure 19), the complexity shifts towards the efficient coverage of multiple targets. The results indicate a significant lead for PreACT over all baselines. By reconciling competing target intents within the same timestep, PreACT agents effectively resolve assignment conflicts, avoiding redundant clustering around single food items and achieving optimal distribution across the map. In contrast, while SeqComm maintains reasonable performance, other baselines such as SMPE, MAPPO, and UDI-Comm struggle to coordinate efficient coverage at this scale. HAPPO, CACL, and MAPPO-IS fail to master the game dynamics, further highlighting the necessity of explicit intra-step coordination for resource allocation.

As illustrated in Figure 20, in the `academy_3_vs_1_with_keeper` scenario, success depends on the precise synchronization of passing and receiving. PreACT and SeqComm achieve comparable leading performance, exhibiting a marginal advantage over UDI-Comm. This suggests that structured coordination is critical for aligning the timing of the passer and the receiver. Standard centralized methods like MAPPO and CACL perform reasonably well but exhibit higher action variance in completing complex pass chains, while HAPPO struggles to stabilize efficient passing policies in this high-dimensional control task.

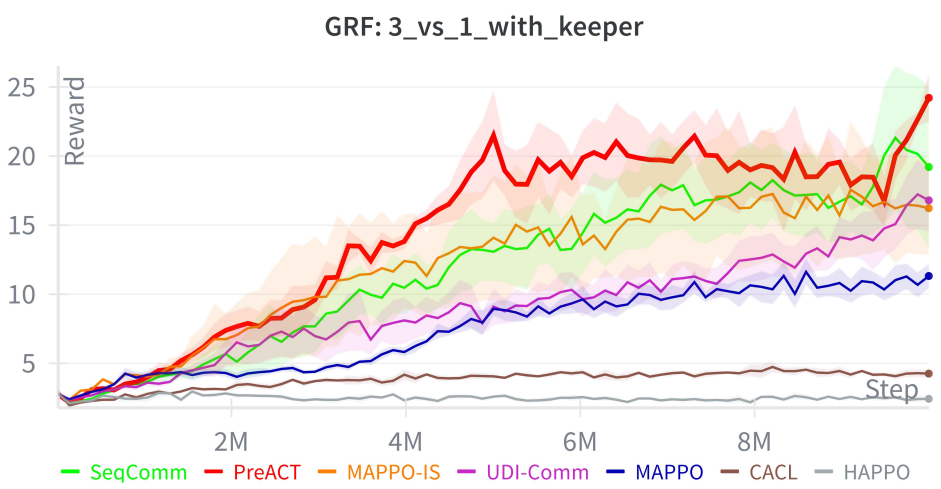


Figure 20: Full Results of `GRF`: `academy_3_vs_1_with_keeper`.

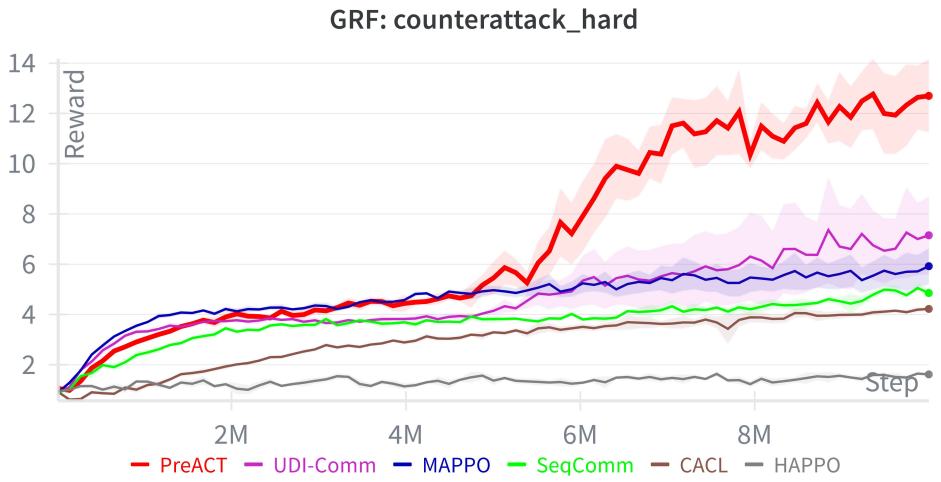


Figure 21: Full Results of `GRF`: `academy_counterattack_hard`.

In the more chaotic `academy_counterattack_hard` scenario (Figure 21), PreACT significantly outperforms all baselines. This task demands the rapid exploitation of a numerical advantage before the defense is organized. PreACT's intra-step coordination proves decisive here, allowing attackers to instantly distribute themselves into distinct lanes to stretch the defensive line. In contrast, baselines such as SeqComm, UDI-Comm, MAPPO, and CACL cluster at a similar performance level, often failing to maximize the spatial advantage, resulting in overcrowded attacks that are easily intercepted.

C.4 Additional Ablation Studies

We conduct further ablation studies to validate the design choices regarding message reconstruction targets and the learning framework.

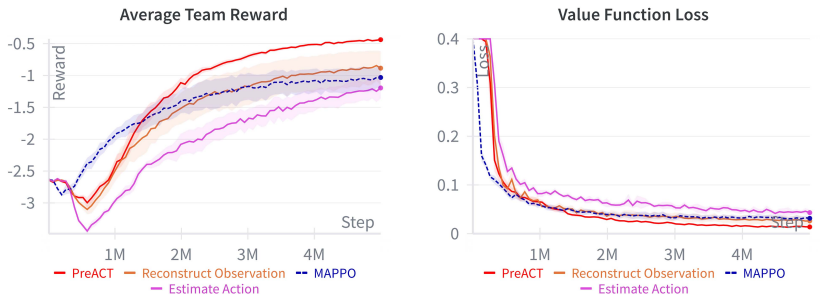


Figure 22: Ablation on Message Reconstruction Targets.

Message Reconstruction Targets. We investigate the impact of the reconstruction target in the self-supervised objective by comparing our full objective (Observation + Action) against variants that reconstruct only the observation or estimate only the draft action. As illustrated in Figure 22, ablating either component leads to performance degradation compared to the full method. Notably, we observe that reconstructing only the observation yields slightly better results than reconstructing only the action. This advantage stems from the fact that incorporating teammates' observations allows the agent to implicitly estimate the global state, thereby mitigating the non-stationarity caused by partial observability, as evidenced by the lower value function loss. However, it is worth noting that the variant utilizing only estimated actions still achieves performance comparable to the centralized MAPPO baseline. This finding is significant because it demonstrates that even within a fully independent learning framework, effective policy coordination via draft actions can bridge the gap to centralized training methods.

Reverse KL Divergence for Draft Action. We further explore the impact of the divergence direction in the message reconstruction loss by comparing our default objective $\mathcal{L}_{act} = D_{KL}(\hat{\pi}_t^i \| \pi_t^i)$ against the reverse KL divergence $D_{KL}(\pi_t^i \| \hat{\pi}_t^i)$. As shown in Figure 23, while the empirical results indicate that both formulations support effective learning and achieve high performance, the default setting demonstrates a slight advantage. This is because the default direction ($D_{KL}(\hat{\pi} \| \pi)$) encourages mode-seeking behavior in the decoder, which forces the message to encode a more decisive draft action by focusing on the peak of the policy distribution. In contrast, the reverse direction tends to be mean-seeking, encouraging the message to cover the entire support of the policy, which may introduce ambiguity into the coordination signal. In the context of intra-step coordination, a firm and unambiguous draft proves more effective for resolving conflicts than a broad distributional estimate.

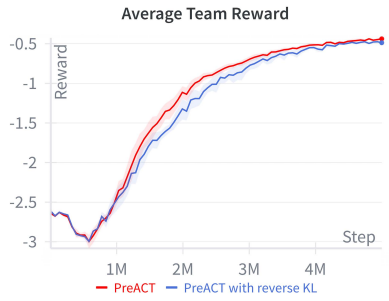


Figure 23: Ablation on reverse KL divergence.

C.5 Computational Overhead and Representation Analysis

In this subsection, we examine the practical implications of the PreACT framework by analyzing its computational overhead and visualizing the semantic structure of the learned communication space.

Computational Overhead Analysis. We provide a detailed analysis of the computational costs associated with PreACT, examining both the inference latency of individual modules and the total training duration.

First, we analyze the micro-level inference latency. Table 3 details the execution time for specific network components across PreACT, MAPPO, and IPPO. Our measurements indicate that the inference cost of PreACT's core policy network ($\mathrm{ObsEnc} + \mathrm{ActGen}$) is strictly comparable to the monolithic policies of MAPPO and IPPO. The additional overhead introduced by the framework primarily arises from the Message Generator (MsgGen), which remains computationally efficient, whereas the Message Decoder (MsgDec) incurs a minimal cost and is utilized solely during the training phase. This breakdown demonstrates that the architectural decomposition introduces negligible latency to the decision-making backbone, ensuring that the communication generation remains highly efficient.

Table 3: Micro-level inference latency per batch (6400 samples). Times are reported in milliseconds (ms). N/A denotes not applicable.

Method	N	**Policy / ActGen**	**Critic**	**MsgGen**	**MsgDec**
IPPO	7	1.654	1.313	N/A	N/A
MAPPO	7	1.724	1.408	N/A	N/A
PreACT (Ours)	7	1.757	1.386	0.567	0.127
IPPO	15	1.669	1.299	N/A	N/A
MAPPO	15	1.933	4.816	N/A	N/A
PreACT (Ours)	15	2.357	2.027	0.593	0.128

Second, we evaluate the macro-level efficiency by measuring the total GPU wall-clock time required to complete one full experimental run under identical conditions (same rollout length, update frequency, and data volume). Table 4 summarizes the results for Team-Reward Cooperative Navigation scenario. While PreACT naturally incurs higher computational costs than non-communicative or simple CTDE baselines like IPPO and MAPPO, it demonstrates superior efficiency compared to other complex coordination frameworks. This places PreACT in a favorable position, offering advanced coordination capabilities with a moderate and scalable computational footprint.

Table 4: Comparison of total training duration in Team-Reward Cooperative Navigation ($N = 7$) scenario. All methods were evaluated on the same hardware.

Method	**Training Time**
CACL	59m
PreACT	**1h 51m**
SMPE	2h 51m
UDI-Comm	2h 54m
MAPPO-IS	3h 16m
SeqComm	4h 33m
IPPO	54m
MAPPO	1h 18m
HAPPO	1h 19m

Message Space Visualization. To verify that the broadcasted messages effectively encode agents' decisions, we employ t-SNE to project the high-dimensional message vectors (m_t^i) into a two-dimensional manifold. Figure 24 visualizes these projections collected from the *Team-Reward Cooperative Navigation* task, colored by the agent's final executed action category. We observe a distinct *hierarchical structure* in the learned manifold. On a macro level, the message space is strictly divided into five separate regions, each corresponding to a specific agent. This separation confirms that the communication protocol inherently preserves the sender identity, which is crucial for the attention mechanism to attribute intent signals correctly. On a micro level, within each agent's region, messages associated with the same executed action form tight, distinct clusters. Furthermore,

the topological arrangement of these action clusters is consistent across all agents. This evidence demonstrates that PreACT successfully learns a structured communication protocol where agents maintain unique identifiable messages while sharing a common semantic syntax for draft actions.

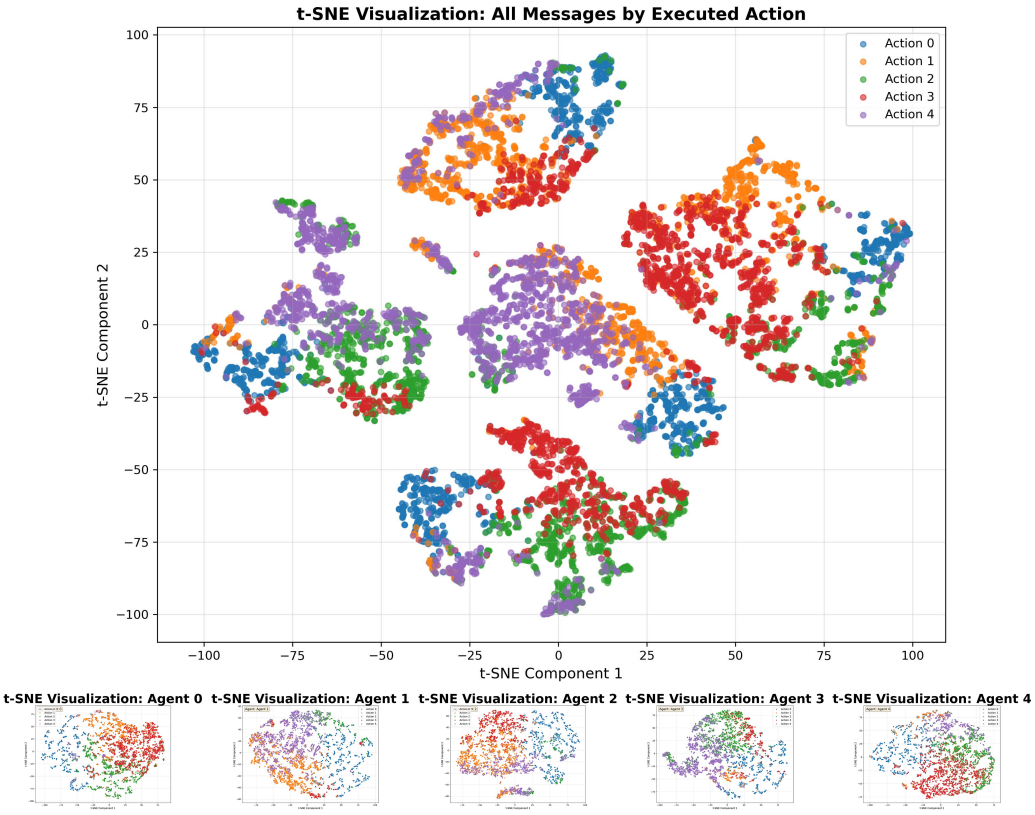


Figure 24: t-SNE visualization of the learned message representations in the *Team-Reward Cooperative Navigation* task ($N = 5$). Points are colored according to the agent's final executed action. **Top:** Aggregated projection of all messages. The space is primarily organized into five distinct macro-regions, each corresponding to a specific agent. Within each agent's region, messages are further clustered by action category, exhibiting a consistent topological arrangement of semantics across the team. **Bottom:** Individual projections for each agent.