Predictive Auxiliary Learning for Belief-based Multi-Agent Systems

Qinwei Huang[1], Simon Khan[2], Rui Zuo[1], Stefan Wang[3], Garrett E. Katz[1], Qinru Qiu[1]
[1]Syracuse University
[2]Air Force Research Laboratory
[3]University of Rochester
{qhuang18, rzuo02, gkatz01, qiqiu}@syr.edu
swang170@u.rochester.edu
simon.khan@us.af.mil

Abstract

Multi-agent reinforcement learning (MARL) under partial observability requires agents to construct reliable belief representations from limited local observations and partial information exchange. Conventional approaches rely primarily on sparse task rewards to shape these internal representations, which often leads to unstable training dynamics and slow convergence. We propose BElief-based Predictive Auxiliary Learning (BEPAL), a decentralized MARL framework that improves belief representation quality through predictive auxiliary tasks. BEPAL trains agents to maintain a "world model" that accurately estimates the global environment state and predicts future dynamics from accumulated local observations and communication, providing another channel of feedback that complements sparse reward in reinforcement learning. By encouraging the hidden state to serve as a compact and informative summary of historical input, BEPAL stabilizes decentralized policy learning and accelerates convergence. The proposed approach is compatible with any decentralized MARL architectures with homogeneous agents and does not increase execution-time complexity. Experiments on Predator–Prey, Traffic Junction, Google Research Football, and RWARE show that BEPAL consistently improves learning stability and task performance compared to strong baselines, highlighting the effectiveness of predictive auxiliary learning for belief formation under partial observability.

Introduction

Cooperative multi-agent systems play a central role in real-world applications such as collaborative robotics, autonomous driving, and distributed sensing. In these settings, agents must coordinate their behaviors to achieve shared objectives while operating under partial observability and limited local information. Due to scalability, privacy, and robustness constraints, centralized control at execution time is often infeasible. As a result, many multi-agent reinforcement learning (MARL) approaches adopt the Centralized Training with Decentralized Execution (CTDE) paradigm, where agents may leverage global information during training but must rely solely on local observations during deployment.

While CTDE improves sample efficiency and training stability, strict implementations often depend on centralized critics that process joint states and actions. Such critics scale poorly with the number of agents and are difficult to apply in large or dynamic systems. Communication-based decentralized MARL frameworks, including IC3Net (Sukhbaatar, Fergus et al. 2016), TarMAC (Das et al. 2019), and related architectures, address this limitation by adopting homogeneous agents and embedding inter-agent communication directly into decentralized policies. These designs preserve scalability while enabling agents to gain global awareness without using a centralized critic.

Despite these advances, decentralized MARL under partial observability remains challenging. Even with inter-agent communication, the information available to each agent is incomplete and fragmented, and must be integrated over time to support effective coordination. In practice, agents rely on a latent internal state that summarizes their observation history and received messages. The quality of this internal representation is critical, as it implicitly determines what information should be retained, propagated, and acted upon. However, in most communication-based MARL methods, this latent state is shaped almost exclusively by sparse and delayed reward signals, which provide weak supervision for learning task-relevant structure. As a result, agents may fail to learn from important but indirectly rewarded information, limiting coordination quality and performance.

This work introduces *BElief-based Predictive Auxiliary Learning* (BEPAL), a decentralized MARL framework designed to strengthen belief representations through predictive auxiliary tasks. BEPAL augments standard communication-based policies with a belief decoder that predicts task-relevant dynamics and estimates unobservable environment information. By comparing these predictions with ground truth, agents receive constant feedback that complements the reward information during training, which guides the hidden state to align better with the latent representation of mission relevant features of the global environment and facilitate the agents to better understand each other's message. Importantly, BEPAL does not add any computational complexity at execution time. It acts as a lightweight and architecture-flexible training assistant and can be integrated into a wide range of communication-based MARL systems.

Through extensive experiments on benchmarks, such as Predator–Prey, Traffic Junction (Sukhbaatar, Fergus et al.

2016), Google Research Football (Kurach et al. 2020), and Robotic Warehouse (Papoudakis et al. 2021), we show that BEPAL consistently improves training stability, accelerates convergence, and enhances final performance across diverse communication structures. Our analysis further demonstrates that a MARL has more performance gain from auxiliary learning if there are less restrictions in inter-agent communication, and that higher belief prediction accuracy is positively correlated to the agents' coordinated performance.

The following summarizes the contributions of this paper:

- We propose BEPAL, a decentralized MARL framework that strengthens latent belief representations through predictive auxiliary learning without adding execution time complexities.
- We show that BEPAL is compatible with a broad class of communication-based MARL architectures trained under CTDE.
- Through extensive experiments and analysis, we demonstrate that the quality of the auxiliary learning of belief prediction positively correlates to the performance improvements of the agents.

Background and Related Work

We investigate a cooperative multi-agent game within the framework of the Decentralized Partially Observable Markov Decision Process (Dec-POMDP), defined as the tuple $\langle N, S, P, \mathcal{R}, \mathcal{O}, \mathcal{A}, Z, \gamma \rangle$. Here, N denotes the number of agents; S is the finite state space; $P(s' \mid s, a) : S \times \mathcal{A} \times S \to [0, 1]$ represents the state transition probabilities; $\mathcal{A} = [\mathbf{A}_1 \dots \mathbf{A}_N]$ constitutes the finite set of actions, with $\mathbf{A}_i$ indicating the local actions $\mathbf{a}_i$ available to agent i; $\mathcal{O} = [\mathbf{O}_1 \dots \mathbf{O}_N]$ encapsulates the finite set of observations governed by the observation function $Z : S \times \mathcal{A} \to \mathcal{O}$; $\mathcal{R} : S \times \mathcal{A} \to \mathbb{R}^N$ is the reward function; and the constant $\gamma \in (0, 1]$ is a discount factor. In each time step t, agent i selects action $\mathbf{a}_i^t$, and receives reward r_i^t and observation o_i^t. Agent i aims to maximize its discounted reward $R_i = \sum_{t=0}^{T} \gamma^t r_i^t$. In Dec-POMDP setting, each agent receives local observations and selects actions to maximize its cumulative reward, while the underlying global state remains unobservable. This formulation captures the uncertainty and distributed nature of real-world multi-agent tasks.

A core challenge in MARL is the non-stationarity induced by multiple agents learning concurrently (Ye, Zhang, and Yang 2015; Xu et al. 2018). Centralized training with decentralized execution (CTDE) mitigates this issue by leveraging joint information during training while maintaining decentralized policies at execution. Methods such as MADDPG (Lowe et al. 2017) and COMA (Foerster et al. 2018) demonstrate improved training stability, but rely on centralized critics whose complexity scales poorly with the number of agents, limiting their applicability in large or dynamic systems. Value decomposition approaches, including VDN (Sunehag et al. 2017), QMIX (Rashid et al. 2020), and QTRAN (Son et al. 2019), improve scalability by factorizing the joint value function into agent-wise components. However, the structural constraints imposed by factorization limit their ability to capture complex inter-agent dependencies, making them less suitable for scenarios where coordination depends on selective communication and belief formation under partial observability.

Communication-based MARL enables agents to exchange information to alleviate partial observability. Early works such as RIAL and DIAL (Foerster et al. 2016) introduced differentiable communication channels, while CommNet (Sukhbaatar, Fergus et al. 2016) proposed mean-based message aggregation that improves coordination but may dilute agent-specific information. IC3Net (Singh, Jain, and Sukhbaatar 2018) added communication gating to decide when agents should broadcast messages, improving efficiency in scenarios with costly communication. More expressive attention-based architectures, including TarMAC (Das et al. 2019) and MAGIC (Niu, Paleja, and Gombolay 2021), selectively weight messages based on relevance and significantly enhance coordination. However, these methods rely solely on sparse reinforcement signals to shape internal representations, making the quality of learned beliefs highly dependent on task rewards.

While communication enables agents to exchange partial observations, several works address partial observability from another direction: learning an internal belief state without relying on any communication. Representative methods such as MA²E (Kang et al. 2025), GAPSR (Zhang et al. 2022) and SIDE (Xu et al. 2021) construct latent belief representations through recurrent encoders, temporal attention, or predictive state estimation. These approaches effectively aggregate historical information and mitigate perceptual aliasing, but they operate strictly under decentralized execution and therefore do not incorporate inter-agent message passing. Moreover, the learned beliefs are typically unstructured latent vectors, lacking the spatial organization required to support explicit reasoning about agent locations, motion patterns, or environmental geometry. As a result, existing belief-based MARL methods either model temporal history without communication, or leverage communication without constructing temporally informed structured beliefs. To the best of our knowledge, no prior work simultaneously integrates temporal modeling, message-based information fusion, and structured map-style belief reconstruction.

Auxiliary learning has been shown to strengthen representation learning in reinforcement learning by providing denser supervision through tasks such as future prediction, reconstruction, or contrastive learning. Works like UNREAL (Jaderberg et al. 2016) and related predictive coding models (Oord, Li, and Vinyals 2018) demonstrate that auxiliary objectives can guide models toward learning meaningful features beyond what is available from the primary reward signal. Applying auxiliary learning to MARL is less common but has gained interest as a means to improve robustness and sample efficiency. For example, BAMS (Luo et al. 2023) incorporates spatial grid maps as predictive targets to refine message content and provide additional training signals. Although grid-map–based supervision improves interpretability and structure, it is limited by the expressiveness of predefined spatial grids and may not generalize well to tasks that require more flexible representation.

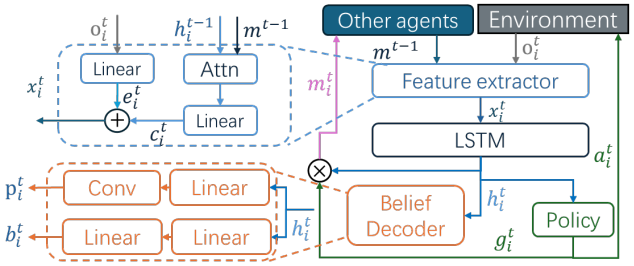

Figure 1: The BEPAL-MAS architecture.

Our work builds on these insights by introducing predictive auxiliary learning designed specifically for communication-based MARL. Rather than imposing rigid spatial priors or relying exclusively on reward-driven learning, BEPAL uses prediction tasks to encourage agents to encode task-relevant motion and local environmental cues in their belief states. This auxiliary supervision complements sparse reinforcement signals and enhances the quality of the internal representations formed from observations and messages. In contrast to prior belief-based methods that construct only latent vectors and operate without communication, BEPAL integrates temporal information carried by message-aware latent features with a structured spatial decoder that reconstructs belief maps directly. This design enables agents to form interpretable, spatially grounded beliefs while leveraging communication for information fusion, filling a gap between communication-focused MARL and temporal belief modeling.

Method

We consider a cooperative multi-agent system with N agents operating under partial observability. From the perspective of each agent, the environment is modeled as a POMDP. At each time step t, agent i receives a local observation o_i^t and inter-agent messages m_{ij}^{t-1} from other agents $j \neq i$. Based on its local observation and received messages, the agent maintains and updates a hidden belief state h_i^t, which serves as a compact internal representation of task-relevant features of the global environment, which is partially observed.

At every step, each agent makes two decisions conditioned on its belief state: a communication action, which determines whether its internal state is broadcast to other agents, and an environment action, which governs how the agent interacts with the environment. To avoid coupling communication control with task execution in a single enlarged action space, we parameterize these decisions using two separate policies, π_θ for communication gating and π_ϕ for environment actions. Both policies share the same belief state h_i^t as input and produce distributions over their respective action spaces. Actions are sampled accordingly, $a_{\theta(\phi)}^t \sim \pi_{\theta(\phi)}(\cdot \mid h_i^t)$.

Beyond action selection, the belief state h_i^t is also used to decode symbolic beliefs b_i^t about unobserved aspects of the global state. The belief decoder is trained to optimize auxiliary objectives, which are application specific. Their ground truth is collected through a centralized process during training, providing constant supervision that helps to shape the internal representation to better capture features in the global environment and facilitate the agents in better understanding each other's message.

The overall BEPAL architecture is illustrated in Figure 1. It consists of a feature extractor, an LSTM-based belief encoder, policy networks for action and communication, and a belief decoder used only during training. We focus on MARL systems with homogeneous agents, where all agents use the same network parameters while conditioning on their own local observations and messages and maintaining a different belief state. This shared-parameter design improves data efficiency and stabilizes training by aggregating experience across agents. The detailed design of each component and the associated training procedure are described in the following subsections.

Feature Extractors

The feature extractor prepares both local observations and incoming messages for belief state updates. Each agent i receives a local observation o_i^t at time t. o_i^t is a vector of concatenated features from processed sensor inputs. BEPAL encodes o_i^t to e_i^t using a fully connected network. This simple architecture was chosen to favor generalizability, as it makes no assumptions about the input format, whether Euclidean space, point clouds, or otherwise. Depending on the sensor modality, application-specific feature extractor, such as a convolutional neural network for image input, may need to be trained end-to-end.

Beyond local observations, agents receive messages from teammates through communication. BEPAL does not assume any specific communication structure, instead, it leverages an attention mechanism to integrate all messages, which are then used to update the beliefs. This design choice makes BEPAL compatible with different communication structures, such as broadcast-based communication or graph-structured message passing, without modifying the core belief learning framework. The agents should communicate with each other not only their current observations, but also their intentions, past experiences, insights derived from those experiences, etc. Hence, each agent sends their hidden state in the message, as it contains comprehensive information about the agent's world model.

During message aggregation, the receiver generates a query vector from its current belief state, while all incoming messages are mapped to key–value pairs through linear projections. The aggregated message context c_i^t is then computed as a weighted combination of message values, where the weights reflect the relevance of that message to the receiver's current belief update.

Hidden State Update and Decision Making

The aggregated observation (e_i^t) and message (c_i^t) are summed and fed into an LSTM network. This LSTM enables the agent to retain historical information, making it easier to learn long-term goals and facilitating better policy learning by compressing task-relevant historical information

into a single vector. The agent's hidden state h_i^t and cell state s_i^{t+1} are updated using Equation 1.

$$h_i^t, s_i^t = lstm(e_i^t + c_i^t, h_i^{t-1}, s_i^{t-1}) \quad (1)$$

Based on the hidden state, the agent chooses actions using a trained policy network within the actor-critic framework. The actor network, denoted as $actor()$, consists of a fully connected linear layer with LogSoftmax activation function. It maps the hidden state to a vector of action probabilities, $\pi(a|h)$, also known as policy. The action taken by agent i at timestamp t, denoted as a_i^t, is sampled from the policy distribution $a_i \sim \pi(a_i^t|h_i^t)$.

At each step, an agent must make two decisions, (1) to select a movement action to advance the game objectives, and (2) to decide whether to broadcast its hidden state. These actions are denoted as a_i^t and g_i^t respectively. The corresponding policies are denoted as π_a and π_g. The outgoing message m_i^t for agent i at timestamp t is the element-wise product of hidden state and binary gate action: $m_i^t = h_i^t \odot g_i^t$.

The critic model, denoted as $critic()$, is a single-layer fully connected network. It gives agent i's local estimation, v_i^t, of the total discounted future rewards.

$$a_i^t \sim \pi(a_i^t \mid h_i^t), \quad g_i^t \sim \pi_g(\cdot \mid h_i^t), \quad v_i^t = critic(h_i^t) \quad (2)$$

where $critic(h_i^t)$ denotes the value estimate used for policy optimization.

BEPAL does not impose any restrictions on communication other than gating at the sender side. Messages are broadcast to all teammates, however, connectivity between agents may be affected by other factors, such as link quality.

Auxiliary Predictive Tasks

In addition to maximizing the policy's reward, the hidden state encoder is trained to optimize some auxiliary objectives via supervised learning. Specifically, a belief decoder is attached to the hidden state, and both encoder and decoder are trained end-to-end to predict task-relevant aspects of the unobserved environment. The auxiliary training does not directly update the actor-critic network. It improves the policy selection by providing a more informative hidden state that accurately represents the global environment.

What environment information should be included in the decoded belief and what network architecture should be chosen for the decoder are task specific. In general, we consider two types of representations of prediction output: vector-based representation and grid-based representation.

Vector-based representations can describe environment features in non-Euclidean spaces, such as an agent's speed, direction, and identities. A two-layer fully connected network is used to decode the agent's belief state h_i^t into a vector representation b_i^t.

Grid-based representation uses voxel grids to describe environment information, such as the location and shape of obstacles, or the distribution of explored areas, etc. A convolutional neural network (CNN) is employed to decode the agent's belief state h_i^t into a grid-based representation p_i^t.

Training the belief encoder and decoder with the auxiliary objectives ensures that the hidden state keeps accurate information about the global environment, including other agents' states, thereby guaranteeing situational awareness and mutual understanding among agents. Importantly, the auxiliary prediction task operates only during training. At execution time, the belief decoder is removed, only the encoder is deployed.

Another benefit of the auxiliary training is that it prevents information loss during policy optimization. Specifically, it encourages the hidden state to retain information about teammates that is critical for coordination but may not directly contribute to immediate rewards. Such information may otherwise be vulnerable to erasure during reward-driven policy optimization, as the primary objective offers no incentive to retain those features.

Loss Functions

We adopt centralized training with decentralized execution, where all agents share the same BEPAL model and are trained jointly. The overall training objective consists of a reinforcement learning loss and an auxiliary loss:

$$L = L_{RL} + \lambda L_{aux},$$

where λ controls the strength of the auxiliary supervision.

The auxiliary loss is defined as the mean squared error between predicted beliefs and their corresponding ground-truth targets:

$$L_{aux} = \sum_{t=1}^{T} \sum_{i=1}^{N} \left(\mu \,\mathrm{MSE}(\bar{b}^t, b_i^t) + \nu \,\mathrm{MSE}(\bar{p}^t, p_i^t) \right), \quad (3)$$

where N is the number of agents and T is the episode length. The coefficients μ and ν balance the contributions of vector-based and grid-based prediction targets, respectively. The auxiliary loss does not directly optimize the policy for better reward. Instead, it regularizes the learning of the hidden state by encouraging it to retain task-relevant latent information that is otherwise weakly supervised by sparse rewards.

The reinforcement learning component follows a standard actor–critic formulation,

$$L_{RL} = L_{actor} + \beta L_{critic},$$

where β balances the actor and critic objectives. The critic loss minimizes the temporal-difference error:

$$L_{critic} = \sum\nolimits_t \left\| r_i^t + \gamma V(h_i^{t+1}) - V(h_i^t) \right\|^2, \quad (4)$$

with $r_i^t = \mathcal{R}(s_i^t, a_i^t)$ denoting the reward received by agent i at time t.

Each agent maintains two actor policies: a game-action policy π_θ and a communication-gating policy π_ϕ. These policies are updated via policy gradient using the critic's value estimates:

$$\nabla_\theta J(\theta) = \sum\nolimits_t \nabla_\theta \log \pi_\theta(a_i^t \mid h_i^t)\, \delta_i^t, \quad (5)$$

$$\nabla_\phi J(\phi) = \sum\nolimits_t \nabla_\phi \log \pi_\phi(g_i^t \mid h_i^t)\, \delta_i^t, \quad (6)$$

where $\delta_i^t = r_i^t + \gamma V(h_i^{t+1}) - V(h_i^t)$ is the temporal-difference advantage.

Gradients from all agents are averaged during training to update the shared model parameters, ensuring consistent representation learning across the system.

Experiments

We evaluate BEPAL's effectiveness within communication-based MARL frameworks where agents broadcast their hidden states. We consider two communication structures: an all-to-all communication and a graph-based communication. For each structure, we compare BEPAL against a canonical baseline: TarMAC for all-to-all communication and MAGIC for graph-structured message-passing.

Evaluation Environments

We evaluate BEPAL on four cooperative multi-agent environments under partial observability: Predator–Prey (PP), Traffic Junction (TJ), Google Research Football (GRF), and Robotic Warehouse (RWARE). These environments feature fundamentally different spatial structures, action-state spaces, and reward functions, enabling us to assess BEPAL's performance across distinct challenges in belief formation, information propagation, and long-horizon coordination.

Predator–Prey (PP) is a grid-based cooperative pursuit task in which $N = 5$ predators operate in a 12×12 map to locate and capture a prey under partial observability. Each predator observes only a local neighborhood and must coordinate with teammates to infer the prey's location and movement. Agents incur a small per-timestep penalty until capture, incentivizing efficient pursuit.

We consider three PP variants with progressively increasing task difficulty and environmental complexity: (1) *stationary prey without obstacles*, where each agent observes only a one-cell neighborhood and captures the prey only when predator and prey occupy the same grid; (2) *stationary prey with 10 obstacles*, which introduces random occlusions constraining navigation while expanding the observation range to two grid cells; and (3) *moving prey with 10 obstacles*, where the prey actively escapes and multiple agents must coordinate to round it up in order for capture. In all variants, the map size and number of agents are fixed, allowing us to isolate the effect of prey dynamics and environmental occlusion on belief accuracy. These PP settings require agents to actively communicate to jointly infer the location and motion of the unobserved prey.

During training, BEPAL agents learn the auxiliary task to predict the locations and motion of other agents and the prey, using a vector-based representation. It also predicts the region that has been explored and the location of obstacles, described using a grid-based representation.

Traffic Junction (TJ) models cooperative navigation under dense and dynamic interactions. Cars are controlled by agents that can see only one grid away and must coordinate via communication to avoid collisions. The agent has two actions (*gas* or *brake*) at each timestep and the goal is to maximize the number of cars reaching destinations.

We evaluate two challenging TJ settings with an 18×18 grid featuring four two-way roads and up to 20 simultaneous agents. The *hard* setting induces high interaction density, while the *hard*+$p_{add} = 0.1$ setting further increases congestion by raising the car arrival rate. These scenarios create frequent occlusions and tightly coupled interactions, where successful coordination depends on accurately anticipating nearby agents' intentions and movements.

During training, BEPAL performs the auxiliary task to predict each car's location, one-hot encoded identity, and next action. All information is represented using a vector-based representation.

Google Research Football (GRF) is a physics-based 3D multi-agent soccer environment with a large action space and stochastic dynamics. We evaluate the academy 3-versus-1 scenario, where three attacking agents face one defender and a goalkeeper controlled by built-in AI. To induce partial observability, each agent's vision is restricted to a fixed radius. As a result, key entities such as the ball and teammates frequently move outside the observation range, requiring agents to maintain latent beliefs over unobserved game states. Episodes terminate when a goal is scored, or the ball goes out of bounds, or possession changes. Performance is measured by the scoring success rate. GRF emphasizes belief persistence under intermittent observation and delayed feedback, as agents must retain information about ball position, player movement, and possession across occlusions.

During training, BEPAL performs the auxiliary task to predict each player and the ball's one-hot encoded identity, position, movement direction, and ball ownership in a vector format. These state predictions provide agents with immediate dynamic context for decision-making.

Robotic Warehouse (RWARE) Robotic Warehouse (RWARE) is a cooperative multi-agent benchmark inspired by automated warehouse systems, where multiple robots must coordinate to deliver loaded shelves to designated goal locations as fast as possible. Agents operate in a grid-based environment with partial observability. Rewards are sparse and delayed. They are received only after a sequence of actions is executed in the correct order: locate the filled shelf, pick up the shelf, go to the station, unload the shelf and put the empty shelf back to a vacant location. As a result, effective performance benefits from agents communicating with each other to exchange information on their route and location of filled/empty shelves.

This sparse reward poses a challenge in MARL, as agents are unlikely to receive any reward through random exploration. To ensure that agents can obtain positive experiences during initial training, we adopt an imitation learning phase in which agents learn from a heuristic teacher model for 6,000 epochs. The teacher is then removed and MARL training continues. For BEPAL, auxiliary learning is performed during both the imitation learning and the subsequent MARL phase.

During training, BEPAL predicts the positions of agents and target shelves as an auxiliary task in a vector format. This supervision encourages the hidden state to retain latent task-level information, such as teammate intent and target occupation, which is difficult to preserve under sparse rewards.

Performance Comparison

Figure 2 and 3 compares the performance of BEPAL with the baseline models across PP, TJ, GRF, and RWARE. As we can see, BEPAL outperforms the baselines in all cases.

Particularly, for RWARE, baseline methods without belief-based prediction collapse rapidly after teacher re-

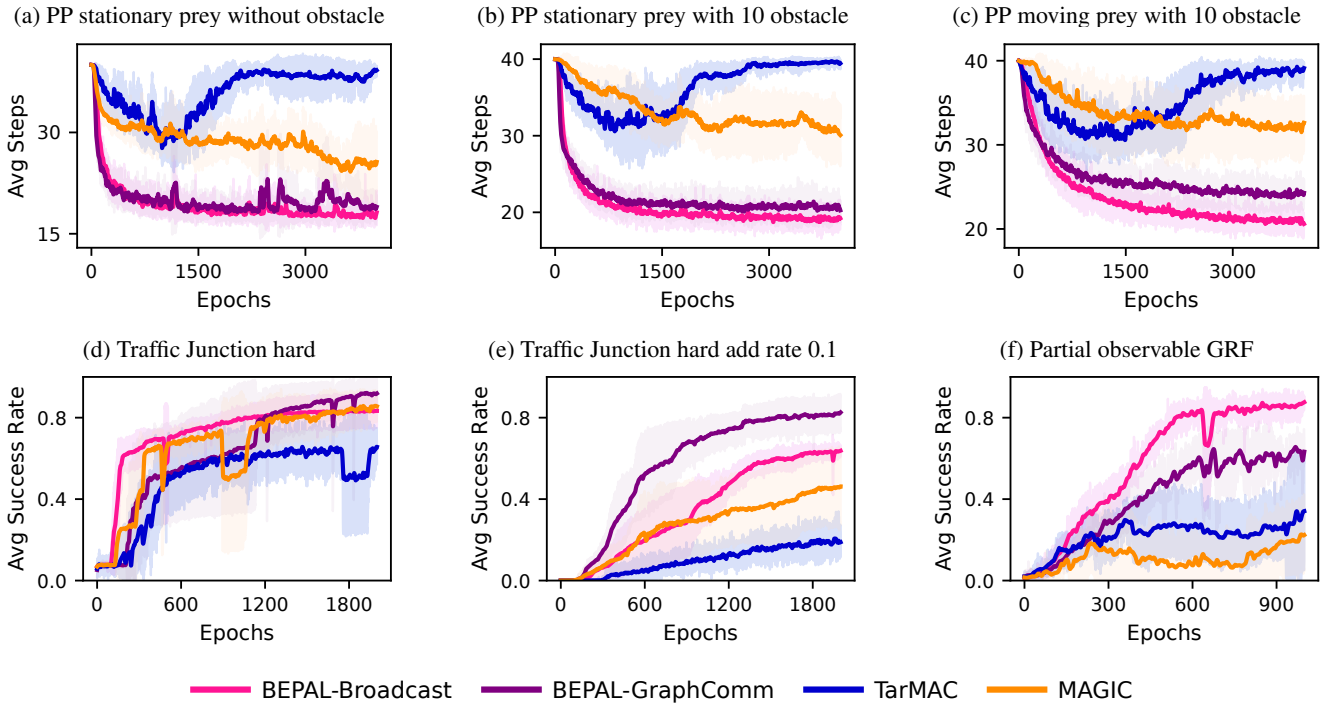

Figure 2: Comparison of belief-based learning under different communication structures and task settings. (a)–(c) PP under three representative settings. (d) TJ (hard). (e) TJ (add rate 0.1). (f) GRF.

Method	PP-without obstacle	PP-with obstacles	PP-moving prey	TJ	TJ-add rate 0.1	GRF	RWARE
Broadcast Comm.	*Avg. steps*	*Avg. steps*	*Avg. steps*	*Success rate*	*Success rate*	*Success rate*	*Reward*
BEPAL-Broadcast	17.65	19.22	20.85	0.8332	0.6375	0.8627	14.11
TarMAC	39.15	39.55	38.93	0.6624	0.1876	0.3080	0.04
Improvement	**21.50**	**20.33**	**18.08**	**0.1708**	**0.4499**	**0.5547**	**14.07**
Graph-based Comm.	*Avg. steps*	*Avg. steps*	*Avg. steps*	*Success rate*	*Success rate*	*Success rate*	*Reward*
BEPAL-GraphComm	18.87	20.59	24.29	0.9200	0.8274	0.6322	13.27
MAGIC	25.39	30.43	32.15	0.8557	0.4632	0.2048	12.29
Improvement	**6.52**	**9.84**	**7.86**	**0.0643**	**0.3642**	**0.4274**	**0.98**

Table 1. Performance and absolute improvements of BEPAL under different communication structures.

moval. TarMAC's performance dropped to near-zero. MAGIC exhibits partial recovery but remains unstable. Both BEPAL-Broadcast and BEPAL-GraphComm have a moderate performance drop after teacher removal and then continue to improve steadily. They eventually outperform the teacher model. This indicates that the BEPAL agents did not blindly mimic the teacher's behavior during the imitation learning. The auxiliary learning enables the agent to better understand the environment. Step-by-step inspection of learned behaviors further shows that BEPAL agents develop coordinated routing strategies that differ qualitatively from the teacher's guidance, such as anticipatory yielding, rerouting in narrow corridors, and implicit role differentiation to avoid deadlocks. These behaviors require maintaining latent beliefs about teammate positions, future motion, and congestion patterns, which cannot be reliably inferred from immediate observations alone.

Impact of Different Communication Protocols

Table 1 summarizes the performance of BEPAL and the baseline models and highlights the agent performance improvement after adopting the auxiliary training. In all environments, incorporating belief-based prediction consistently improves agent performance compared to the corresponding non-belief baselines.

Two interesting observations can be made. First, different types of game have a preference for different communication structure. For example, PP, GRF and RWARE favor the broadcasting communication and the TJ game favors the graph-based communication. This is because agents in games such as PP need to collaborate with each other to locate and capture the prey. However, in a TJ game, agents

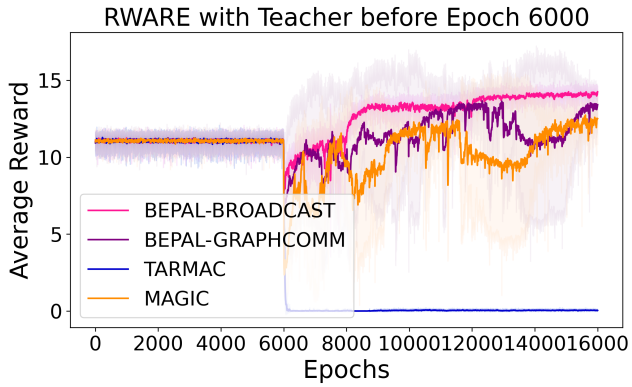

Figure 3: RWARE with teacher-assisted warm-up. Shaded regions indicate standard deviation across runs.

only need to coordinate with their near neighbors to avoid collision.

The second observation we can make is that applying BEPAL in a broadcasting communication environment always leads to more improvement than applying it in a graph based communication environment. In graph-based message passing, far-away agents communicate with each other through multi-hop communication. Agents usually do not possess the most up-to-date information of the global environment. Due to the lack of information, even with the help of auxiliary learning, agents do not have good global situation awareness. In contrast, agents under broadcast communication have real-time information of the global environment. Hence, they could benefit more from the auxiliary learning.

This perspective provides a coherent explanation for why different environments exhibit different improvement patterns. In PP and GRF, where successful coordination often relies on anticipating global motion patterns (e.g., prey escape trajectories or ball dynamics under occlusion), limitations in global information propagation tend to constrain the achievable belief accuracy. In TJ, where coordination is dominated by dense but local interactions (e.g., queue formation and nearby vehicle motion), localized relational reasoning is more informative, and structured communication can provide high-quality local context that auxiliary prediction can exploit effectively. In all cases, the key point is not that a particular architecture is universally superior, but that the communication structure determines the ceiling on information quality, which then bounds belief accuracy and the eventual extent of performance improvement.

Evolution of Agent Beliefs: a Case Study

We use an example scenario of PP(3) to illustrate how belief representations evolve over time and how these evolving beliefs shape coordinated agent behavior.

Figure 4 visualizes belief states of agent 2 collected at three representative timesteps (steps 3, 5, and 7) in a moving prey episode. At step 3, no predator has directly observed the prey. Predator 2 therefore maintains a highly inaccurate belief prediction, with the predicted prey position biased toward unexplored regions of the map. This prediction is driven by an exploration-driven hypothesis rather than real sensory input, indicating that the agent actively maintains an estimation of unknown area even in the absence of direct evidence. At step 5, Predator 1 already observed the prey and broadcasts this information through communication. Although Predator 2 still does not observe the prey directly, its belief state is immediately updated. The predicted prey position shifts toward the true location, and the predicted motion aligns with the prey's escape direction. This demonstrates that BEPAL effectively integrates communicated information into its belief representation, allowing agents to infer latent dynamics beyond their local field of view. By step 7, Predator 2's belief has largely converged. The predicted prey position closely matches the ground truth, and the predicted motion accurately reflects the prey's trajectory. At this stage, the belief state provides sufficiently precise localization to support coordinated pursuit.

Overall, this case study shows that BEPAL's belief representations are not auxiliary artifacts but actionable internal models of the environment. By progressively refining latent prey estimates through communication-driven belief updates, BEPAL enables faster information propagation and more effective multi-agent coordination.

	TJ Hard	**TJ Add Rate 0.1**	**GRF**
Pearson Corr	-0.25	-0.36	-0.40
Spearman Corr	-0.54	-0.86	-0.94

Table 2. Correlation between belief prediction error and performance across environments.

Correlation between Belief Accuracy and Performance

Figure 5 visualizes this relationship between the accuracy of the auxiliary prediction and the agent performance for the game in the PP environment. Each point corresponds to a single training episode, where belief accuracy is measured by the decoder loss and performance is measured by the average number of steps required to capture the prey. Similar information is reported in Table 2 for TJ and GRF.

Across all settings, we observe that the belief prediction error has a strong positive correlation with the number of steps needed to complete the game or a negative correlation with the game performance. Lower prediction error is consistently associated with faster prey capture, indicating that more informative belief states enable more effective coordination and decision-making. This trend holds across different task configurations. The results confirm our hypothesis that the auxiliary training helps the MARL agents to perform better.

Robustness and Transferability

To evaluate transferability, we first train a model on a 12×12 map with 5 agents and 10 obstacles with stationary prey, then transfer it to different test configurations. Results in Table 3 show that the transferred model remains competitive across all settings and, in the large-map configuration

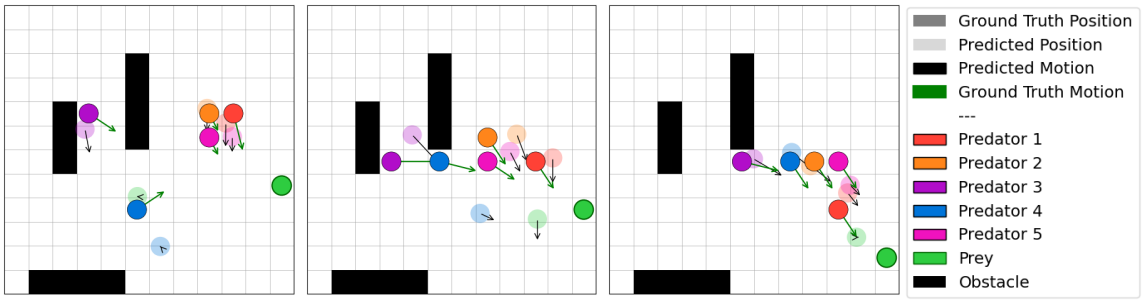

Figure 4: Evolution of belief predictions from the perspective of Predator 2 in the PP environment. From left to right: step 3, step 5 , and step 7.

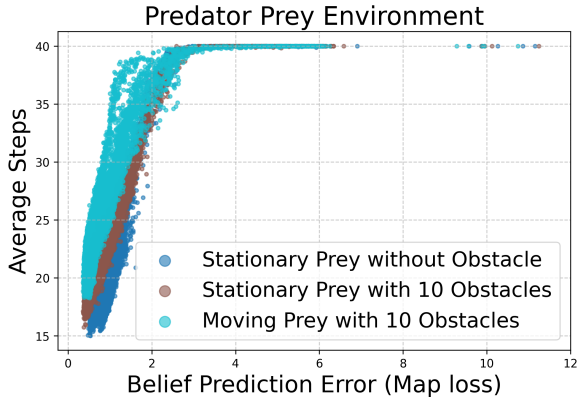

Figure 5: Relationship between belief prediction error and performance in PP.

	N=5, m=12 (0 obs, T=40)	**N=5, m=12** (10 obs, T=40)	**N=5, m=20** (20 obs, T=80)
Native	18.23	22.41	55.23
Transfer	19.76	22.41	54.62

Table 3. Transferability study in the PP environment (average steps to capture the prey).

($m = 20$), even surpasses the native model trained directly on the target environment. This suggests that auxiliary prediction encourages beliefs that capture transferable dynamics, rather than overfitting to a specific layout.

Conclusions

We propose BElief-based Predictive Auxiliary Learning (BEPAL), a decentralized MARL framework that improves belief representation quality under partial observability through predictive auxiliary learning. By training agents to predict global state information and future dynamics from accumulated local observations and communication, BEPAL provides an additional learning signal beyond sparse task rewards, encouraging compact and informative hidden states. Extensive experiments on Predator–Prey, Traffic Junction, Google Research Football, and RWARE show that BEPAL consistently improves learning stability and task performance across different communication protocols, highlighting the importance of belief representation quality for effective multi-agent coordination.

Acknowledgments

This research is partially supported by the Air Force Office of Scientific Research (AFOSR), under contract FA9550-24-1-0078, and NSF award CNS-2148253. The paper was received and approved for public release by Air Force Research Laboratory (AFRL) on case number AFRL-2026-0968. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFRL or its contractors.

References

Das, A.; Gervet, T.; Romoff, J.; Batra, D.; Parikh, D.; Rabbat, M.; and Pineau, J. 2019. Tarmac: Targeted multi-agent communication. In *International Conference on machine learning*, 1538–1546. PMLR.

Foerster, J.; Assael, I. A.; De Freitas, N.; and Whiteson, S. 2016. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29.

Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Jaderberg, M.; Mnih, V.; Czarnecki, W. M.; Schaul, T.; Leibo, J. Z.; Silver, D.; and Kavukcuoglu, K. 2016. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*.

Kang, S.; Lee, Y.; Kim, G.; Chong, S.; and Yun, S.-Y. 2025. MA^{2}E: Addressing Partial Observability in Multi-Agent Reinforcement Learning with Masked Auto-Encoder. In *The Thirteenth International Conference on Learning Representations*.

Kurach, K.; Raichuk, A.; Stańczyk, P.; Zajac, M.; Bachem, O.; Espeholt, L.; Riquelme, C.; Vincent, D.; Michalski, M.; Bousquet, O.; et al. 2020. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 4501–4510.

Lowe, R.; Wu, Y. I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30.

Luo, C.; Huang, Q.; Wu, A. B.; Khan, S.; Li, H.; and Qiu, Q. 2023. Multi-Agent Cooperative Games Using Belief Map Assisted Training. In *ECAI 2023*, 1617–1624. IOS Press.

Niu, Y.; Paleja, R. R.; and Gombolay, M. C. 2021. Multi-Agent Graph-Attention Communication and Teaming. In *AAMAS*, volume 21, 20th.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Papoudakis, G.; Christianos, F.; Schäfer, L.; and Albrecht, S. V. 2021. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative Tasks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS)*.

Rashid, T.; Samvelyan, M.; De Witt, C. S.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178): 1–51.

Singh, A.; Jain, T.; and Sukhbaatar, S. 2018. Learning when to communicate at scale in multiagent cooperative and competitive tasks. *arXiv preprint arXiv:1812.09755*.

Son, K.; Kim, D.; Kang, W. J.; Hostallero, D. E.; and Yi, Y. 2019. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, 5887–5896. PMLR.

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

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J. Z.; Tuyls, K.; et al. 2017. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*.

Xu, Z.; Bai, Y.; Li, D.; Zhang, B.; and Fan, G. 2021. Side: State inference for partially observable cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2105.06228*.

Xu, Z.; Lyu, Y.; Pan, Q.; Hu, J.; Zhao, C.; and Liu, S. 2018. Multi-vehicle flocking control with deep deterministic policy gradient method. In *2018 IEEE 14th International Conference on Control and Automation (ICCA)*, 306–311. IEEE.

Ye, D.; Zhang, M.; and Yang, Y. 2015. A multi-agent framework for packet routing in wireless sensor networks. *sensors*, 15(5): 10026–10047.

Zhang, Z.; Yang, Z.; Liu, H.; Tokekar, P.; and Huang, F. 2022. Reinforcement learning under a multi-agent predictive state representation model: Method and theory. In *The Tenth International Conference on Learning Representations (ICLR 2022)*.