BEHAVIOUR-AWARE DELAY-ALIGNED BELIEF FOR MULTI-AGENT REINFORCEMENT LEARNING WITH COMMUNICATION DELAYS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent reinforcement learning with communication delays is challenging as information transmission among agents is not always instantaneous: each agent acts on local states that are stale by a different lag on each connection. State-of-the-art (SOTA) methods typically employ a belief representation that first predicts the current global state, and the agent then learns and acts on that predicted state. We theoretically demonstrate that the prediction error of such a belief is governed by two quantities: the approximation errors of the dynamics and of the agents' behaviours within the delay window, both amplified and accumulated with the delay by recursive forward propagation. We further analyse how belief error affects the induced policy and value function. Guided by these results, we propose Behaviour-Aware Delay-Aligned Belief (BADA). Specifically, BADA formulates the ragged information state into one token per other agent, aligned to that agent's freshest received state, and directly forecasts the agents' behaviours and the current global state explicitly. The empirical results show that BADA can effectively reduce the belief error compared to SOTA baselines. Furthermore, BADA achieves superior task performance in various benchmarks under different delays.

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) has been applied to a wide range of tasks, such as multi-robot coordination, traffic signal control, power dispatch and real-time strategy games (Oliehoek et al., 2016; Zhang et al., 2021a). Value-factorized MARL methods support centralized training with decentralized execution, where each agent acts on its own information (Sunehag et al., 2017; Rashid et al., 2020). We study a communication-enabled setting in which agents additionally receive state information from one another. In real-world applications, this communicated information has to travel over a physical transmission channel, for example a radio channel, a vehicle-to-vehicle hop or a congested network, and can therefore arrive late (Olfati-Saber & Murray, 2004; Guo et al., 2023; Ma et al., 2024). Moreover, the connections are not alike: each carries a delay of its own, and each delay varies over time, so that an agent holds a ragged information state in which every other agent is stale by a different timestep. Such communication delays can impair coordination and reduce task performance in MARL (Song et al., 2025; Fu et al., 2025).

Communication delays can make the received observation non-Markovian: it combines information from earlier states rather than revealing the current state. In single-agent delayed RL, classical approaches augment delayed observations with the intervening actions to recover a sufficient statistic, at the cost of an input space that grows with the delay (Katsikopoulos & Engelbrecht, 2003; Walsh et al., 2009; Ramstedt & Pal, 2019; Bouteiller et al., 2021). Belief-based approaches instead predict the current state to mitigate delays and support decision-making (Wu et al., 2024; 2025; Tu et al., 2026). In the single-agent setting, the intervening actions are the agent's own and available to itself. In multi-agent systems, however, state reconstruction must also account for other agents' unobserved actions. This raises the question: **how can an agent predict the current global state under heterogeneous communication delays without observing other agents' intervening actions?**

We begin with a theoretical analysis of the belief error bound between exact and approximated beliefs over the delay window. Under common Lipschitz continuity assumptions, the error bound

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.

combines transition and behaviour approximation errors with a propagation factor determined by the delay and the smoothness of transitions and policies. Behaviour approximation can therefore contribute to the bound even when the dynamics are known exactly. We then derive the difference bound between induced policies, establish the Lipschitz continuity of value functions, and analyse joint value function differences. These results relate belief accuracy to policy and value function properties, motivating explicit behaviour modelling alongside state reconstruction.

Guided by this analysis, we propose Behaviour-Aware Delay-Aligned Belief (BADA). BADA formulates the available information into one token per other agent, aligned to its freshest received state, and uses a shared two-head transformer (Vaswani et al., 2017) to predict behaviour and reconstruct the current state in a single forward pass. Specifically, the behaviour head predicts agents' behaviours over the delay window. These predicted behaviours will be fed back to the state head, which predicts agents' current states in the same forward pass without intermediate state rollouts. Each agent combines these predictions with its own current state and supplies the reconstruction to its QMIX utility. The empirical results on various benchmarks show that BADA outperforms existing methods in terms of belief accuracy and task performance under constant and random delays.

Our contributions are summarized as follows:

- We theoretically derive the belief error bound through transition and behaviour approximation errors and analyse its effect on induced policies and joint value functions.
- We propose BADA, which combines delay-aligned representations with behaviour feedback to reconstruct the current state in a single forward pass.
- We evaluate BADA under different delays, demonstrating improved belief accuracy and superior task performance across multiple benchmarks.

2 PRELIMINARIES

Markov game. We consider a Markov game (Zhang et al., 2021a) $\langle \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \{\mathcal{R}^i\}_{i\in\mathcal{N}}, \gamma\rangle$, where $\mathcal{N} = \{1, \dots, N\}$, $\mathcal{S} = \prod_i \mathcal{S}^i$, $\mathcal{A} = \prod_i \mathcal{A}^i$, and $\gamma \in [0, 1)$. The transition dynamics are given by $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Omega(\mathcal{S})$, where $\Omega(\mathcal{S})$ denotes probability measures on $\mathcal{S}$, and agent i receives $r_t^i = \mathcal{R}^i(s_t, a_t)$. We consider that next-state components are conditionally independent given the current state and joint action, adopting factored dynamics (Guestrin et al., 2001): $\mathcal{P}(s_{t+1}|s_t, a_t) = \prod_i \mathcal{P}^i(s_{t+1}^i|s_t, a_t)$. We consider a stationary joint policy with conditionally independent actions given the state, $\pi(a|s) = \prod_i \pi^i(a^i|s)$. The value function is $V_\pi^i(s) := \mathbb{E}_{\pi,\mathcal{P}}[\sum_{t\geq 0} \gamma^t r_t^i \mid s_0 = s]$, and the Q-function is $Q_\pi^i(s, a^i) := \mathbb{E}_{\pi,\mathcal{P}}[\sum_{t\geq 0} \gamma^t r_t^i \mid s_0 = s, a_0^i = a^i]$. We use $-i$ to denote the set of all agents except agent i (e.g., a^{-i} and π^{-i}).

QMIX. QMIX (Rashid et al., 2020) represents the joint action value as $Q^{\text{total}}(s, a) := f\big(s, (Q^i(\cdot, a^i))_{i\in\mathcal{N}}\big)$, where f is conditioned on the global state and monotone in each utility. Each Q^i uses the agent's available information. QMIX minimizes the square of the temporal-difference (TD) residual $r_t + \gamma \max_{a'} Q_{\text{target}}^{\text{total}}(s_{t+1}, a') - Q^{\text{total}}(s_t, a_t)$, where r_t is the reward and $Q_{\text{target}}^{\text{total}}$ denotes the target network.

Lipschitz continuity. We assume compact metric spaces $(\mathcal{S}^i, d_{\mathcal{S}^i})$ and $(\mathcal{A}^i, d_{\mathcal{A}^i})$, with additive product metrics $d_{\mathcal{S}}(s_1, s_2) = \sum_i d_{\mathcal{S}^i}(s_1^i, s_2^i)$ and $d_{\mathcal{A}}(a_1, a_2) = \sum_i d_{\mathcal{A}^i}(a_1^i, a_2^i)$. The 1-Wasserstein distance $\mathcal{W}$ uses the metric of the underlying space.

Definition 1 (Lipschitz continuity (Rachelson & Lagoudakis, 2010))**.** The transition, rewards, and policy are Lipschitz continuous if $L_{\mathcal{P}^i}$, $L_{\mathcal{R}^i}$, and L_{π^i} satisfy, for all $(s_1, a_1), (s_2, a_2) \in \mathcal{S} \times \mathcal{A}$,

$$\mathcal{W}\big(\mathcal{P}^i(\cdot|s_1, a_1) \| \mathcal{P}^i(\cdot|s_2, a_2)\big) \leq L_{\mathcal{P}^i}\big(d_{\mathcal{S}}(s_1, s_2) + d_{\mathcal{A}}(a_1, a_2)\big),$$
$$|\mathcal{R}^i(s_1, a_1) - \mathcal{R}^i(s_2, a_2)| \leq L_{\mathcal{R}^i}\big(d_{\mathcal{S}}(s_1, s_2) + d_{\mathcal{A}}(a_1, a_2)\big),$$
$$\mathcal{W}\big(\pi^i(\cdot|s_1) \| \pi^i(\cdot|s_2)\big) \leq L_{\pi^i} d_{\mathcal{S}}(s_1, s_2).$$

These conditions (Definition 1) are commonly adopted in the Lipschitz-MDP literature (Asadi et al., 2018; Liotet et al., 2022). Under the transition factorization and additive metric, a product coupling

gives $\mathcal{W}(\mathcal{P}(\cdot|s_1,a_1)\|\mathcal{P}(\cdot|s_2,a_2)) \le L_{\mathcal{P}}(d_{\mathcal{S}}(s_1,s_2)+d_{\mathcal{A}}(a_1,a_2))$, where $L_{\mathcal{P}} := \sum_i L_{\mathcal{P}^i}$ is a valid upper bound on the joint transition's Lipschitz constant.

3 Problem Setting

We consider a Markov game under communication delays (Fu et al., 2025). At timestep t, let $\delta_t^{i,j} \in \{0,\dots,\Delta\}$ denote the age of the latest state of agent j available to agent i, where Δ is the maximum delay. Each agent observes its own current state, so $\delta_t^{i,i} = 0$. For agent i, the latest received states are $o_t^i := \big(s^1_{t-\delta_t^{i,1}},\dots,s^N_{t-\delta_t^{i,N}}\big)$. Let $\delta_t^i := \max_j \delta_t^{i,j}$. We assume that agent i has buffered the complete state history of each agent from $t-\delta_t^i$ through that agent's latest available timestep, together with its own actions ($a^i_{t-\delta_t^i:t-1}$). Therefore, the information available to agent i over this delay window is

$$x_t^i := \big(s^1_{t-\delta_t^i:t-\delta_t^{i,1}},\dots,s^N_{t-\delta_t^i:t-\delta_t^{i,N}},a^i_{t-\delta_t^i:t-1}\big). \tag{1}$$

Equation (1) extends the augmentation technique used in delayed RL (Katsikopoulos & Engelbrecht, 2003; Fu et al., 2025). Other agents' intervening actions are unavailable at execution ($a^{-i}_{t-\delta_t^i:t-1}$), so predicting their current states requires accounting for these unobserved actions.

For the error analysis, we align all agents to the same timestep $t-\delta_t^i$. The history assumption makes the common snapshot $s_{t-\delta_t^i}$ available. Both rollouts start from the same timestep and use the observer's historical actions as inputs; other agents' actions follow the exact or approximated behaviour model. Therefore, the fundamental challenge of MARL with communication delays is how to model the behaviour information from the other agents, which is not directly observable, and how to incorporate the behaviour information into belief prediction. Formally, the belief representation is defined as:

$$b^{(k+1)}_{\phi,\psi}(\cdot|x_t^i) := \int \mathcal{P}_\phi\big(\cdot \mid s,(a^i_{t-\delta_t^i+k},a^{-i})\big)\pi_\psi^{-i}(\mathrm{d}a^{-i}|s)\, b^{(k)}_{\phi,\psi}(\mathrm{d}s|x_t^i), \quad k=0,\dots,\delta_t^i-1, \tag{2}$$

where $\mathcal{P}_\phi(s'|s,a) = \prod_j \mathcal{P}^j_\phi(s'^j|s,a)$ and $\pi^{-i}_\psi(a^{-i}|s) = \prod_{j\neq i}\pi^j_\psi(a^j|s)$, parameterized by ϕ and ψ, respectively. Replacing these two approximations with exact $\mathcal{P}$ and the policy π^{-i} defines the exact belief $b^{(k+1)}$. We write the final belief predictions at $k=\delta_t^i$ as $b_{\phi,\psi}(\cdot|x_t^i)$ and $b(\cdot|x_t^i)$. Both beliefs propagate from the common snapshot without conditioning on newer observations in x_t^i. The approximation errors of the transition and policy are defined in the following Definition 2.

Definition 2 (Approximation Errors)**.** The approximation error bounds of the transition and policy are respectively defined as follows:

$$\epsilon^i_{\mathcal{P}} := \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{W}\big(\mathcal{P}^i(\cdot|s,a)\big\|\mathcal{P}^i_\phi(\cdot|s,a)\big), \quad \epsilon^i_\pi := \sup_{s\in\mathcal{S}} \mathcal{W}\big(\pi^i(\cdot|s)\big\|\pi^i_\psi(\cdot|s)\big).$$

4 Theoretical Analysis

In this section, we analyse how transition and behaviour approximation errors affect belief prediction under communication delays (Theorem 1) and propagate to policy action distributions (Corollary 1). We then establish the regularity of fixed-policy value functions (Proposition 1) and bound belief-induced changes in value-factorization outputs (Corollary 2). All proofs are provided in Appendix A.

Belief Error Analysis. We first bound the belief difference between the exact and approximated beliefs in the following theorem.

Theorem 1 (Belief Error Bound)**.** *Given x_t^i, the approximated belief error is bounded by*

$$\mathcal{W}\big(b(\cdot|x_t^i)\big\|b_{\phi,\psi}(\cdot|x_t^i)\big) \le \underbrace{\left(\sum_{i\in\mathcal{N}}\epsilon^i_{\mathcal{P}} + L_{\mathcal{P}}\sum_{j\neq i}\epsilon^j_\pi\right)\frac{1-(L_{\mathcal{P}}(1+\sum_{j\neq i}L_{\pi^j}))^{\delta_t^i}}{1-L_{\mathcal{P}}(1+\sum_{j\neq i}L_{\pi^j})}}_{\epsilon^i_b}$$

Theorem 1 identifies two sources of the belief error: transition approximation and prediction of other agents' unobserved behaviour. Even with exact dynamics ($\epsilon^i_{\mathcal{P}} = 0$), behaviour error contributes through the transition smoothness $L_{\mathcal{P}}$. The propagation ratio $L_{\mathcal{P}}(1 + \sum_{j\neq i} L_{\pi^j})$ determines how these errors accumulate and amplify.

Policy Difference Analysis. We next show how belief error affects policies. Given the policy π^i, we define $\pi^i(\cdot|\beta) := \int \pi^i(\cdot|s)\,\beta(\mathrm{d}s)$, where $\beta \in \{b(\cdot|x^i_t), b_{\phi,\psi}(\cdot|x^i_t)\}$. The following corollary bounds the difference between the policies induced by the exact and approximated beliefs.

Corollary 1 (Joint Policy Difference)**.** *Given policy π^i and x^i_t, we have*

$$\mathcal{W}\big(\pi^i(\cdot|b(\cdot|x^i_t))\big\|\ \pi^i(\cdot|b_{\phi,\psi}(\cdot|x^i_t))\big)\ \le\ L_{\pi^i}\epsilon^i_b.$$

For $\pi_{\phi,\psi} = (\pi^i_{\phi,\psi})_{i\in\mathcal{N}}$ and $\pi_b = (\pi^i_b)_{i\in\mathcal{N}}$, whose agents respectively act on $b_{\phi,\psi}$ and b, we have

$$\mathcal{W}\big(\pi_b(\cdot|x_t)\big\|\ \pi_{\phi,\psi}(\cdot|x_t)\big)\ \le\ \sum_{i\in\mathcal{N}} L_{\pi^i}\epsilon^i_b.$$

Corollary 1 shows that belief error can affect the belief-induced policies, thus affecting the behaviour of the agents.

Value Difference Analysis. We next analyse how belief error affects value estimates. We first establish the Lipschitz continuity of value functions under a fixed policy. Proposition 1 relates value smoothness to the smoothness of rewards, transitions, and policies.

Proposition 1 (Lipschitz Continuous Q^i_π and V^i_π)**.** *For a fixed stationary joint policy π satisfying Definition 1, if $\gamma L_{\mathcal{P}}\big(1+\sum_{j\neq i} L_{\pi^j}\big)\big(1+L_{\pi^i}\big) < 1$, then Q^i_π and V^i_π are Lipschitz continuous under $d_{\mathcal{S}} + d_{\mathcal{A}^i}$ and $d_{\mathcal{S}}$, respectively, with valid constants*

$$L_{Q^i} = \frac{L_{\mathcal{R}^i}(1+\sum_{j\neq i} L_{\pi^j})}{1-\gamma L_{\mathcal{P}}(1+\sum_{j\neq i} L_{\pi^j})(1+L_{\pi^i})}, \qquad L_{V^i} = L_{Q^i}(1+L_{\pi^i}).$$

We now consider joint action values formed by a mixing network. Since learned utilities need not equal Q^i_π, we state their regularity separately.

Assumption 1 (Utility and mixer regularity)**.** For each agent i, the fixed utility satisfies $|Q^i(s_1,a^i) - Q^i(s_2,a^i)| \le L_{Q^i} d_{\mathcal{S}}(s_1,s_2)$ uniformly over $a^i \in \mathcal{A}^i$. The fixed mixer is monotone in each utility and satisfies $|f(s,u) - f(s,u')| \le L_f\|u-u'\|_1$ uniformly over its state input s, with finite constants L_{Q^i} and L_f.

Monotonicity supports decentralized greedy maximization (Rashid et al., 2020), while L_f bounds the mixer's sensitivity to utility errors. For an additive mixer, $L_f = 1$. For fixed network parameters, mixer state s_t, and joint action a_t, we define the joint action value under belief β as

$$Q^{\mathrm{total}}_\beta(s_t,a_t) := f\Big(s_t, \big(\mathbb{E}_{s\sim\beta(\cdot|x^i_t)}\big[Q^i(s,a^i_t)\big]\big)_{i\in\mathcal{N}}\Big), \qquad \beta \in \{b,\ b_{\phi,\psi}\}. \tag{3}$$

The following corollary bounds the value difference induced by the exact and approximated beliefs.

Corollary 2 (Joint Action Value Difference)**.** *Under Assumption 1, for fixed x_t, s_t and any joint action $a_t \in \mathcal{A}$, the outputs in Equation* (3) *satisfy*

$$\big|Q^{\mathrm{total}}_b(s_t,a_t) - Q^{\mathrm{total}}_{b_{\phi,\psi}}(s_t,a_t)\big| \le L_f \sum_{i\in\mathcal{N}} L_{Q^i}\epsilon^i_b.$$

Corollary 2 shows how individual belief errors contribute to the joint value difference. Each error is weighted by the corresponding utility sensitivity, and the mixer scales their combined effect. Thus, reducing belief error tightens the value-difference bound, especially for agents with more sensitive utilities.

Remark 1 (Single-agent Generalization)**.** Our theoretical results (Theorem 1 and Proposition 1) can generalize to the single-agent setting. For the single-agent setting ($N = 1$), ϵ^i_b reduces to $\epsilon^i_{\mathcal{P}}(1 - L^{\delta^i_t}_{\mathcal{P}})/(1 - L_{\mathcal{P}})$, the single-agent compounding-error form of Wu et al. (2025). Similarly, the Q-function constant in Proposition 1 reduces to $L_{\mathcal{R}}/\big(1 - \gamma L_{\mathcal{P}}(1 + L_\pi)\big)$ in the single-agent setting (Rachelson & Lagoudakis, 2010).

5 BADA: BEHAVIOUR-AWARE DELAY-ALIGNED BELIEF

Theorem 1 identifies two sources of the belief error: transition and behaviour approximation errors. It also shows how successive predictions propagate these errors across the delay window. This suggests predicting behaviour explicitly while reconstructing the current state. Guided by these observations, we propose Behaviour-Aware Delay-Aligned Belief (BADA), shown in Figure 1. BADA combines delay-aligned tokens with a shared two-head transformer inspired by Wu et al. (2024) and adopts direct state prediction (Fu et al., 2025; Wu et al., 2025). We introduce behaviour feedback: the behaviour head predicts other agents' action distributions, which the state head uses to forecast their current states. The theoretical result motivates these prediction targets and the use of available information.

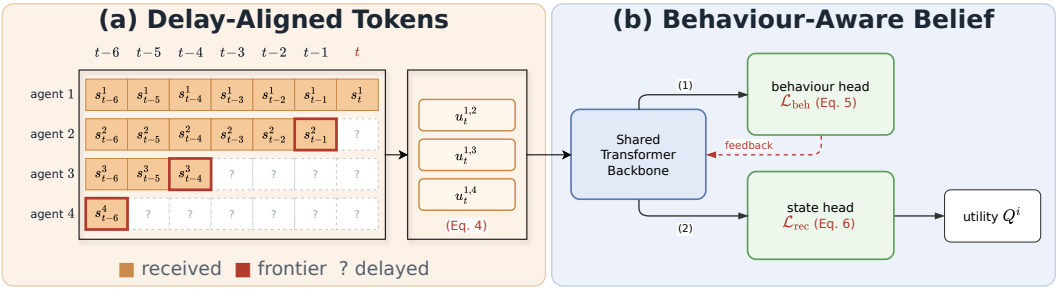

Figure 1: Architecture of BADA. (a) Delay-aligned tokens of Equation (4). (b) The two-head transformer, predicting the action distributions of the other agents with the behaviour head and then the current global state with the state head.

Delay-aligned representation. We retain each agent's recent observations by serializing x_t^i into one token per other agent, aligned to its freshest received state $s^j_{t-\delta_t^{i,j}}$:

$$u_t^j := \left(s^j_{t-\delta_t^i:\, t-\delta_t^{i,j}},\ s^i_{t-\delta_t^i:\, t},\ a^i_{t-\delta_t^i:\, t-1},\ \delta_t^{i,j}\right), \qquad j \neq i. \tag{4}$$

Each token records the received information, the observer's own state and action histories, and the delay. This preserves the distinction between known observations and the unobserved evolution beyond each agent's frontier.

Behaviour Prediction and State Reconstruction. The behaviour error term in Theorem 1 motivates explicit supervision of other agents' action distributions. Given $(u_t^j)_{j\neq i}$, the behaviour head $\hat{\pi}_\psi$, parameterized by ψ, predicts action distributions for each other agent and timestep within the delay window. Specifically, the behaviour head is optimized by the cross-entropy loss:

$$\mathcal{L}_{\text{beh}}(\psi) = \mathbb{E}\Big[\text{CE}\big(\hat{\pi}_\psi(\cdot|(u_t^j)_{j\neq i}), (\pi^j(\cdot|s_k)_{t-\delta_t^i \le k<t})_{j\neq i}\big)\Big]. \tag{5}$$

In this loss, $\pi^j(\cdot|s_k)$ denotes the supervision reference recomputed at timestep k from agent j's delay-free local observation. It need not match the distribution used during delayed execution and the implementation details are detailed in Appendix B.2.

The state head $\hat{b}_\phi$, parameterized by ϕ, receives the predicted behaviour as feedback. This behaviour feedback conditions reconstruction on estimates of the actions. The head directly predicts $(s_t^j)_{j\neq i}$ in the same forward pass, avoiding recursive forward prediction over the delay window. The state head is then optimized by the reconstruction loss:

$$\mathcal{L}_{\text{rec}}(\phi) = \mathbb{E}\Big[\text{MSE}\big(\hat{b}_\phi((u_t^j, \hat{a}^j_{t-\delta_t^i:\, t-1})_{j\neq i}), (s_t^j)_{j\neq i}\big)\Big], \tag{6}$$

where $\hat{a}^j_{t-\delta_t^i:\, t-1}$ is the sequence of full action-probability vectors for agent j, and MSE is the mean squared error loss. The two heads are trained jointly with total loss:

$$\mathcal{L}(\phi, \psi) = \lambda_{\text{beh}}\mathcal{L}_{\text{beh}}(\psi) + \lambda_{\text{rec}}\mathcal{L}_{\text{rec}}(\phi), \tag{7}$$

where λ_{beh} and λ_{rec} respectively weight behaviour prediction and state reconstruction. We set $\lambda_{\text{beh}} = 1$ and $\lambda_{\text{rec}} = 3$ by default and provide ablation studies in Section 6.4.

Integration with QMIX. We integrate BADA with QMIX by supplying reconstructed states to the agents' utilities. Agent i combines its predicted states with its own current state to form $\hat{s}_t$ and selects actions using $Q^i(\hat{s}_t, a^i)$. For each training batch, we first update BADA, then update QMIX by minimizing the TD residual with reconstructions. The bounds in Corollaries 1 and 2 motivate accurate reconstruction by relating belief error to policy and value differences. In the next section, we empirically evaluate BADA's belief accuracy and task performance under various delays.

6 Experimental Results

6.1 Experiment settings

We evaluate BADA on MPE (Lowe et al., 2017) and SMAX with SMACv2-style scenarios under both constant and random delays, using JaxMARL (Rutherford et al., 2024). We compare BADA against delay-aware QMIX (Delay-aware), Recursive Belief, RDC-GRU, RDC-TF (Fu et al., 2025), DDC (Mednikov & Gal, 2026), DFBT (Wu et al., 2025), and DUPO (Tu et al., 2026). Specifically, we adapt the single-agent delayed RL methods DFBT and DUPO to our multi-agent setting. All methods use QMIX as the algorithm backbone. Each method is run over 10 random seeds with 10M environment steps per seed. We follow the delay protocol of Fu et al. (2025), and provide implementation details and hyper-parameters in Appendix B. We provide additional empirical results on classic SMAX, Hanabi (Bard et al., 2020), and Coin Game (Raileanu et al., 2018) in Appendix C.

6.2 Belief error comparison

Following the offline belief comparison in Wu et al. (2025), we evaluate all belief methods on a shared dataset. For our MPE experiments, this dataset is collected during delay-free QMIX training. Figure 2 compares belief error across different delays and numbers of agents. BADA consistently achieves the lowest belief error across the evaluated settings, maintaining its advantage as delays and team sizes increase. These results demonstrate the effectiveness of combining delay-aligned representations with behaviour feedback for direct state reconstruction. They also provide empirical support for the theoretical motivation in Theorem 1, which highlights behaviour approximation and error propagation as key considerations in belief prediction under communication delays.

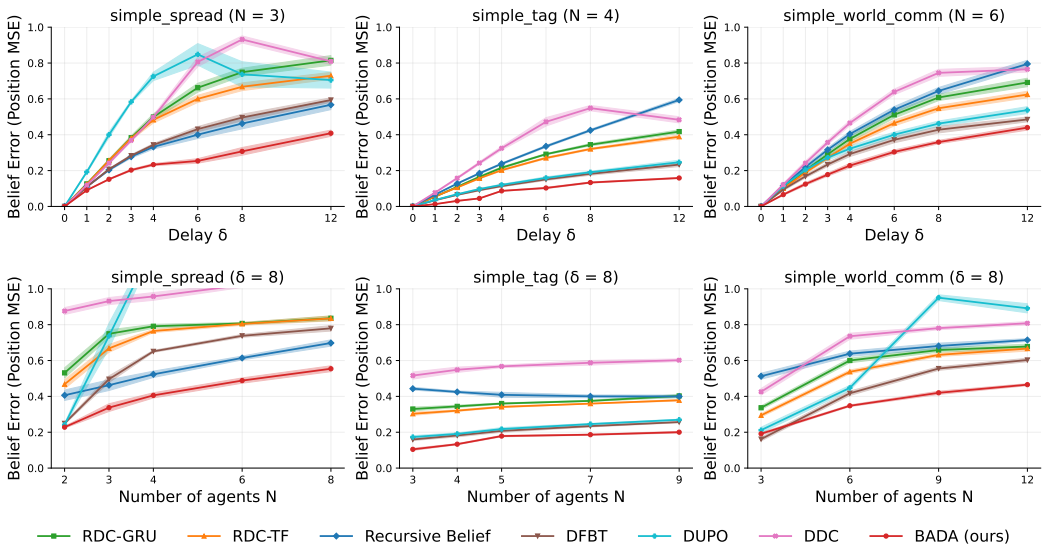

Figure 2: Belief error comparison with different delays δ (top) and number of agents N (bottom).

6.3 PERFORMANCE COMPARISON

As shown in Figures 3 and 4, we report the performance of BADA and baselines on MPE and SMAX tasks under different delays.

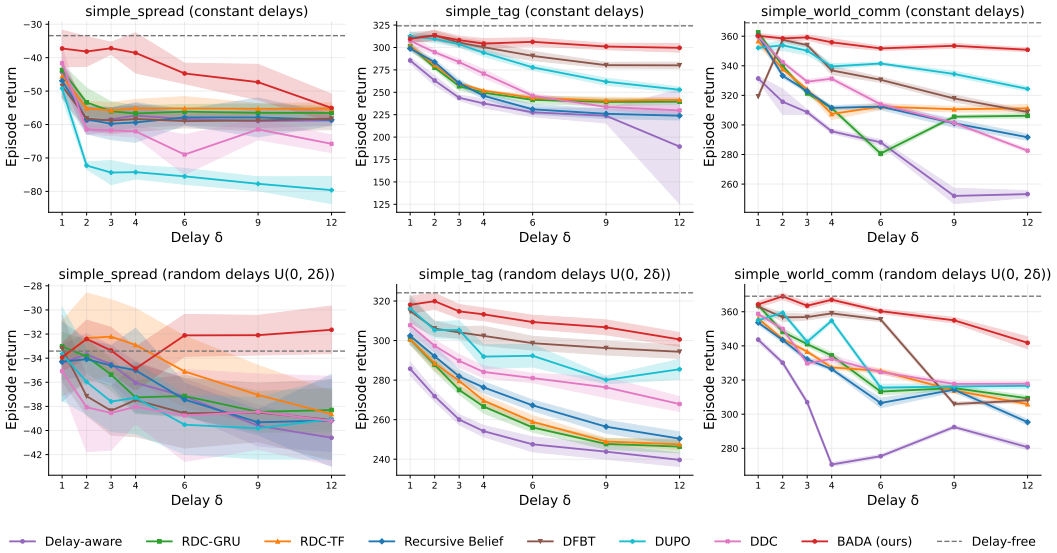

Figure 3: Performance on MPE with constant delays (top) and random delays (bottom).

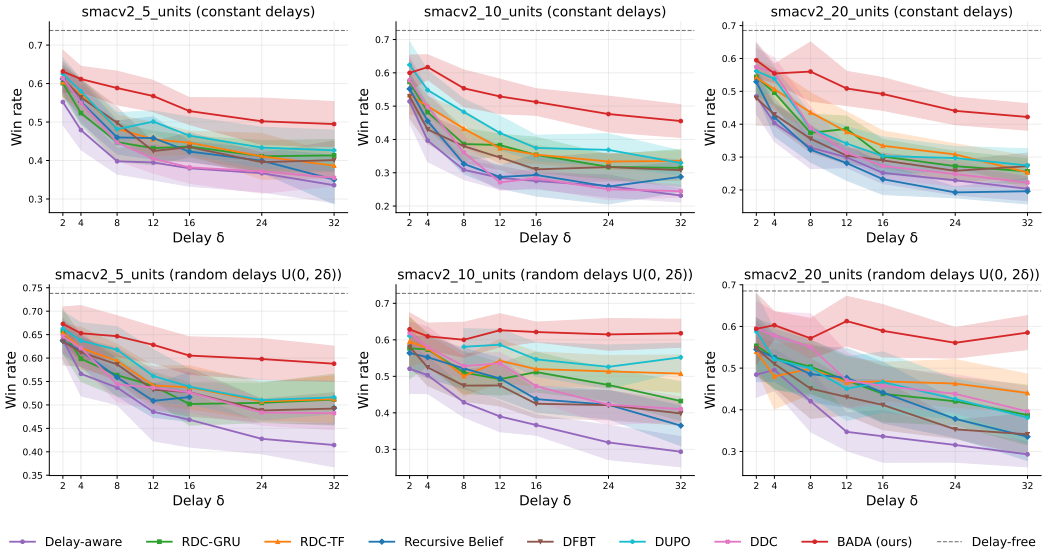

Figure 4: Performance on SMAX with constant delays (top) and random delays (bottom).

Constant delays. BADA achieves superior performance on both SMAX and MPE tasks under constant delays. On SMAX, several baselines remain competitive at short delays, while BADA's advantage becomes more pronounced at larger delays, particularly on smacv2_10_units and smacv2_20_units. On MPE, BADA maintains high returns on simple_tag and simple_world_comm as delays increase. Although its return on simple_spread declines at larger delays, it remains competitive with the strongest baselines. These results demonstrate BADA's effectiveness when agents must act on increasingly stale information.

Random delays. BADA also performs well when communication delays become random, achieving the highest performance in most settings. On the larger SMAX tasks, its win rates remain relatively stable across the tested delay ranges, while most baselines deteriorate. On MPE, BADA maintains strong performance on simple_tag and simple_world_comm and shows a clearer advantage on simple_spread at larger random delays. This supports the use of delay-aligned representations and behaviour feedback under heterogeneous, time-varying communication delays.

6.4 Ablation study

We evaluate the behaviour module and loss weights on MPE simple_spread with random delays $U(0, 12)$ and SMAX smacv2_5_units with random delays $U(0, 16)$. Table 1 reports belief error and task performance for each variant, with all variants trained separately under the same experimental protocol as BADA.

	simple_spread		smacv2_5_units	
Method	Belief Error	Return	Belief Error	Win Rate
BADA ($\lambda_{\text{beh}} = 1$, $\lambda_{\text{rec}} = 3$)	0.26 ± 0.01	-32.11 ± 1.74	2.92 ± 0.04	0.65 ± 0.04
w/o behaviour head and feedback	0.37 ± 0.01	-41.88 ± 1.06	4.95 ± 0.06	0.49 ± 0.05
w/o behaviour feedback	0.32 ± 0.03	-38.88 ± 1.00	4.75 ± 0.03	0.53 ± 0.03
w/ oracle behaviour feedback	0.20 ± 0.01	-31.43 ± 1.48	2.29 ± 0.03	0.69 ± 0.04
$\lambda_{\text{beh}} = 0.3$, $\lambda_{\text{rec}} = 3$	0.30 ± 0.01	-32.49 ± 0.71	4.23 ± 0.04	0.52 ± 0.05
$\lambda_{\text{beh}} = 3$, $\lambda_{\text{rec}} = 3$	0.27 ± 0.02	-33.82 ± 3.11	3.73 ± 0.03	0.60 ± 0.04
$\lambda_{\text{beh}} = 1$, $\lambda_{\text{rec}} = 1$	0.31 ± 0.02	-33.49 ± 3.11	3.82 ± 0.04	0.62 ± 0.03
$\lambda_{\text{beh}} = 1$, $\lambda_{\text{rec}} = 10$	0.30 ± 0.01	-34.79 ± 3.11	3.97 ± 0.04	0.62 ± 0.04

Table 1: Ablation studies of BADA on simple_spread and smacv2_5_units.

Behaviour module. Compared with removing both the behaviour head and feedback, BADA reduces belief error by 29.7% on simple_spread and 41.0% on smacv2_5_units, while improving return from -41.88 to -32.11 and win rate from 0.49 to 0.65. Relative to retaining behaviour-head training without feedback, BADA reduces error by 18.8% and 38.5%, respectively, and improves both performance metrics. This supports directly conditioning reconstruction on predicted behaviour beyond using behaviour supervision alone. Oracle behaviour feedback, which supplies privileged executed actions as feedback, further reduces error by 23.1% and 21.6%, but yields smaller control gains, reaching a return of -31.43 and a win rate of 0.69.

Loss weights. We vary one loss weight at a time while keeping the other at its default. Among the tested configurations, $(\lambda_{\text{beh}}, \lambda_{\text{rec}}) = (1, 3)$ achieves the lowest mean belief error and highest mean task performance on both tasks. Increasing or decreasing either weight worsens reconstruction, with larger relative changes on SMAX. These results support jointly balancing behaviour supervision and state reconstruction, rather than placing greater emphasis on either objective alone.

7 Related work

Markov property does not hold when introducing delays, and the existing method augments the delayed observation with the actions taken since it was produced, which restores a sufficient statistic (Katsikopoulos & Engelbrecht, 2003; Walsh et al., 2009; Ramstedt & Pal, 2019; Bouteiller et al., 2021). Other approaches address delays through delay-free behaviour imitation or learned belief prediction: DIDA (Liotet et al., 2022) imitates a delay-free expert, VDPO (Wu et al., 2024) casts delayed RL as variational inference and reads the belief out of a two-head transformer, DFBT (Wu et al., 2025) forecasts the belief in one shot rather than recursively, and DUPO (Tu et al., 2026) uses a diffusion model to estimate the current-state distribution and weights policy updates by action-value uncertainty. In these single-agent settings, the intervening actions are the learner's own and are available to it.

In our multi-agent setting, an agent observes its own intervening actions but not those of its teammates. Predicting their current states from delayed information therefore requires accounting for these unobserved actions. DAMARL (Chen et al., 2020) handles action delays by augmenting the state with pending actions, while RDC (Fu et al., 2025) compensates for observation delays by reconstructing current observations from observation and action histories together with delay information. DDC (Mednikov & Gal, 2026) filters the delayed observation through learned dynamics to protect a pre-trained policy, and model-based MARL rolls a learned model forward while predicting the other agents along the way (Zhang et al., 2021b). For recursive prediction, model errors can accumulate over the delay window, while errors in predicting unobserved teammate actions introduce an additional source of belief error, as characterized by Theorem 1. Communication-focused approaches learn how long to wait for messages (Yuan et al., 2023), align intent representations and message timeliness (Song et al., 2025), and distinguish communication gains from delay costs (Gao et al., 2026). Complementary theoretical work studies structural equivalence and differences in learning dynamics between observation and action delays (Sintes et al., 2026). Control-theoretic studies analyse consensus under communication delays (Olfati-Saber & Murray, 2004; Badran et al., 2025) and establish safety guarantees for systems with state delays (Kiss et al., 2023).

8 Limitations and challenges

Further theoretical analysis. We analyze the belief error and its effects on belief-induced policy and value functions. Extending these results to sample complexity and convergence remains challenging. During learning, changes in the belief training process affect action selection and the resulting trajectories, while policy updates change the behaviour that the belief must model. This coupling makes it difficult to separate prediction error from policy optimization and distribution shift. We will explore theoretical tools for analysing this joint learning process and connecting belief accuracy to control performance.

More challenging delay settings. Our experiments consider constant and uniformly sampled random delays. Temporally correlated delays, prolonged communication interruptions, and delays that depend on agents' behaviour introduce more difficult prediction settings. Such conditions can leave agents without recent information for extended periods and make information availability depend on their decisions. Future work will evaluate BADA under these conditions to examine its robustness and identify when state reconstruction requires additional communication or adaptation.

Improving behaviour prediction and feedback. Oracle behaviour feedback further reduces belief error, indicating the potential benefit of more informative feedback. Realizing this benefit is challenging because executed actions depend on agents' information, which the observer cannot fully recover from its delayed history. We will investigate richer temporal behaviour models and uncertainty-aware feedback to improve how the state head uses inaccurate predictions. The oracle results motivate these improvements, while the privileged information they use means that the full oracle gain may remain unattainable at execution.

9 Conclusion

This work investigates MARL under communication delays, where agents predict the current state from stale information without observing other agents' intervening actions. Our theoretical analysis bounds belief error through transition and behaviour approximation errors accumulated over the delay window, and further relates belief error to differences in belief-induced policy and value functions. Guided by these results, we propose BADA, a belief architecture that combines delay-aligned representations with explicit behaviour prediction. BADA feeds predicted action distributions into state reconstruction, recovering an estimate of the current global state in a single forward pass without intermediate state rollouts. Experiments demonstrate that BADA effectively reduces belief error and improves task performance across multiple benchmarks under constant and random delays. Ablation studies further confirm the contribution of behaviour feedback to both reconstruction accuracy and performance metrics. Meanwhile, we also acknowledge that there is further room for improvement in behaviour prediction and feedback, and this work provides a theoretical foundation for future research on MARL under communication delays.

AI USE STATEMENT

AI tools were used solely to assist with language editing, grammar checking, and manuscript review. All research ideas, methodological development, theoretical analysis, experimental design and implementation, interpretation of results, and scientific conclusions were developed and carried out by the human authors without the use of AI.

ETHICS STATEMENT

This work does not involve human subjects or sensitive data, and all experiments were conducted in accordance with the principles of responsible research.

REPRODUCIBILITY STATEMENT

We provide detailed proofs of our theoretical results in Appendix A and implementation details, including experimental settings and hyper-parameters, in Appendix B. The code is included in the supplementary material to support reproduction of our experiments.

REFERENCES

Kavosh Asadi, Dipendra Misra, and Michael Littman. Lipschitz continuity in model-based reinforcement learning. In *International conference on machine learning*, pp. 264–273. PMLR, 2018.

Layan A Badran, Kiarash Ayrankia, and Rastko R Selmic. Stability and convergence analysis of multi-agent consensus with communication delays: A lambert w function approach, 2025.

Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

Yann Bouteiller, Simon Ramstedt, Giovanni Beltrame, Christopher J Pal, and Jonathan Binas. Reinforcement learning with random delays. In *ICLR*, 2021.

Baiming Chen, Mengdi Xu, Zuxin Liu, Liang Li, and Ding Zhao. Delay-aware multi-agent reinforcement learning for cooperative and competitive environments, 2020.

Songchen Fu, Siang Chen, Shaojing Zhao, Letian Bai, Ta Li, and Yonghong Yan. Rainbow delay compensation: a multi-agent reinforcement learning framework for mitigating delayed observation, 2025.

Zihong Gao, Hongjian Liang, Lei Hao, and Liangjun Ke. Communication gain and delay cost under cross-timestep delays in cooperative multi-agent reinforcement learning, 2026.

Carlos Guestrin, Daphne Koller, and Ronald Parr. Multiagent planning with factored mdps. volume 14, 2001.

Guodong Guo, Mengfan Zhang, Yanfeng Gong, and Qianwen Xu. Safe multi-agent deep reinforcement learning for real-time decentralized control of inverter based renewable energy resources considering communication delay. *Applied energy*, 349:121648, 2023.

Konstantinos V Katsikopoulos and Sascha E Engelbrecht. Markov decision processes with delays and asynchronous cost collection. *IEEE transactions on automatic control*, 48(4):568–574, 2003.

Adam K Kiss, Tamas G Molnar, Aaron D Ames, and Gabor Orosz. Control barrier functionals: Safety-critical control for time delay systems. *International journal of robust and nonlinear control*, 33(12):7282–7309, 2023.

Pierre Liotet, Davide Maran, Lorenzo Bisi, and Marcello Restelli. Delayed reinforcement learning by imitation. In *International conference on machine learning*, pp. 13528–13556. PMLR, 2022.

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.

Guoqi Ma, Prabhakar R Pagilla, and Swaroop Darbha. Benefits of v2v communication in connected and autonomous vehicles in the presence of delays in communicated signals, 2024.

Maxim Mednikov and Oren Gal. Decoupled delay compensation: Enhancing pre-trained marl policies via learned dynamics filtering, 2026.

Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control*, 49(9):1520–1533, 2004.

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

Emmanuel Rachelson and Michail G Lagoudakis. On the locality of action domination in sequential decision making. In *ISAIM*, 2010.

Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. Modeling others using oneself in multi-agent reinforcement learning. In *International conference on machine learning*, pp. 4257–4266. PMLR, 2018.

Simon Ramstedt and Chris Pal. Real-time reinforcement learning. volume 32, 2019.

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. volume 21, pp. 1–51, 2020.

Alexander Rutherford, Benjamin Ellis, Matteo Gallici, Jonathan Cook, Andrei Lupu, Garðar Ingvarsson, Timon Willi, Ravi Hammond, Akbir Khan, Christian S de Witt, et al. Jaxmarl: Multi-agent rl environments and algorithms in jax. volume 37, pp. 50925–50951, 2024.

Jules Sintes, Ana Bušić, and Jiamin Zhu. Structural equivalence and learning dynamics in delayed marl, 2026.

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

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. 2017.

Junqi Tu, Zejiao Liu, Fangfei Li, and Yang Tang. Diffusion-guided uncertainty-aware delayed policy optimization, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. volume 30, 2017.

Thomas J Walsh, Ali Nouri, Lihong Li, and Michael L Littman. Learning and planning in environments with delayed feedback. *Autonomous Agents and Multi-Agent Systems*, 18(1):83–105, 2009.

Qingyuan Wu, Simon S Zhan, Yixuan Wang, Yuhui Wang, Chung-Wei Lin, Chen Lv, Qi Zhu, and Chao Huang. Variational delayed policy optimization. volume 37, pp. 54330–54356, 2024.

Qingyuan Wu, Yuhui Wang, Simon Sinong Zhan, Yixuan Wang, Chung-Wei Lin, Chen Lv, Qi Zhu, Jürgen Schmidhuber, and Chao Huang. Directly forecasting belief for reinforcement learning with delays, 2025.

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

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms, 2021a.

Weinan Zhang, Xihuai Wang, Jian Shen, and Ming Zhou. Model-based multi-agent policy optimization with adaptive opponent-wise rollouts. 2021b.

A PROOFS

Theorem 1 (Belief Error Bound). *Given x_t^i, the approximated belief error is bounded by*

$$\mathcal{W}\big(b(\cdot|x_t^i)\big\|b_{\phi,\psi}(\cdot|x_t^i)\big) \le \underbrace{\left(\sum_{i\in\mathcal{N}}\epsilon_{\mathcal{P}}^i + L_{\mathcal{P}}\sum_{j\neq i}\epsilon_\pi^j\right)\frac{1-(L_{\mathcal{P}}(1+\sum_{j\neq i}L_{\pi^j}))^{\delta_t^i}}{1-L_{\mathcal{P}}(1+\sum_{j\neq i}L_{\pi^j})}}_{\epsilon_b^i}$$

Proof. Let $\mathbb{E}$ denote the expectation under the optimal coupling of $b^{(\delta_t^i-1)}(\cdot|x_t^i)$ and $b_{\phi,\psi}^{(\delta_t^i-1)}(\cdot|x_t^i)$, writing (s_1, s_2) for the coupled pair.

$$\begin{aligned}
&\mathcal{W}\big(b(\cdot|x_t^i)\big\|b_{\phi,\psi}(\cdot|x_t^i)\big)\\
=&\mathcal{W}\big(b^{(\delta_t^i)}(\cdot|x_t^i)\big\|b_{\phi,\psi}^{(\delta_t^i)}(\cdot|x_t^i)\big)\\
=&\mathcal{W}\Big(\mathcal{P}\big(\cdot\,\big|b^{(\delta_t^i-1)}(\cdot|x_t^i),a_{t-1}^i,\pi^{-i}(\cdot|s_{t-1})\big)\Big\|\mathcal{P}_\phi\big(\cdot\,\big|b_{\phi,\psi}^{(\delta_t^i-1)}(\cdot|x_t^i),a_{t-1}^i,\pi_\psi^{-i}(\cdot|s_{t-1})\big)\Big)\\
\le&\mathcal{W}\Big(\mathcal{P}\big(\cdot\,\big|b^{(\delta_t^i-1)}(\cdot|x_t^i),a_{t-1}^i,\pi^{-i}(\cdot|s_{t-1})\big)\Big\|\,\mathcal{P}\big(\cdot\,\big|b_{\phi,\psi}^{(\delta_t^i-1)}(\cdot|x_t^i),a_{t-1}^i,\pi_\psi^{-i}(\cdot|s_{t-1})\big)\Big)\\
&+\mathcal{W}\Big(\mathcal{P}\big(\cdot\,\big|b_{\phi,\psi}^{(\delta_t^i-1)}(\cdot|x_t^i),a_{t-1}^i,\pi_\psi^{-i}(\cdot|s_{t-1})\big)\Big\|\,\mathcal{P}_\phi\big(\cdot\,\big|b_{\phi,\psi}^{(\delta_t^i-1)}(\cdot|x_t^i),a_{t-1}^i,\pi_\psi^{-i}(\cdot|s_{t-1})\big)\Big)\\
\le&\mathbb{E}\Big[\mathcal{W}\Big(\mathcal{P}\big(\cdot\,\big|s_1,a_{t-1}^i,\pi^{-i}(\cdot|s_1)\big)\Big\|\,\mathcal{P}\big(\cdot\,\big|\,s_2,\,a_{t-1}^i,\,\pi_\psi^{-i}(\cdot|s_2)\big)\big)\Big)\Big]+\sum_{i\in\mathcal{N}}\epsilon_{\mathcal{P}}^i\\
\le&\mathbb{E}\Big[L_{\mathcal{P}}\Big(d_{\mathcal{S}}(s_1,s_2)+\mathcal{W}\big(\pi^{-i}(\cdot|s_1)\big\|\pi_\psi^{-i}(\cdot|s_2)\big)\Big)\Big]+\sum_{i\in\mathcal{N}}\epsilon_{\mathcal{P}}^i\\
\le&\mathbb{E}\Big[L_{\mathcal{P}}\Big(d_{\mathcal{S}}(s_1,s_2)+\mathcal{W}\big(\pi^{-i}(\cdot|s_1)\big\|\pi^{-i}(\cdot|s_2)\big)+\mathcal{W}\big(\pi^{-i}(\cdot|s_2)\big\|\pi_\psi^{-i}(\cdot|s_2)\big)\Big)\Big]+\sum_{i\in\mathcal{N}}\epsilon_{\mathcal{P}}^i\\
\le&\mathbb{E}\Big[L_{\mathcal{P}}\Big(d_{\mathcal{S}}(s_1,s_2)+\sum_{j\neq i}L_{\pi^j}d_{\mathcal{S}}(s_1,s_2)+\sum_{j\neq i}\epsilon_\pi^j\Big)\Big]+\sum_{i\in\mathcal{N}}\epsilon_{\mathcal{P}}^i\\
=&\sum_{i\in\mathcal{N}}\epsilon_{\mathcal{P}}^i+L_{\mathcal{P}}\sum_{j\neq i}\epsilon_\pi^j+L_{\mathcal{P}}(1+\sum_{j\neq i}L_{\pi^j})\mathcal{W}\big(b^{(\delta_t^i-1)}(\cdot|x_t^i)\big\|b_{\phi,\psi}^{(\delta_t^i-1)}(\cdot|x_t^i)\big)\\
\le&(\sum_{i\in\mathcal{N}}\epsilon_{\mathcal{P}}^i+L_{\mathcal{P}}\sum_{j\neq i}\epsilon_\pi^j)\big(1+L_{\mathcal{P}}(1+\sum_{j\neq i}L_{\pi^j})\big)\\
&+\big(L_{\mathcal{P}}(1+\sum_{j\neq i}L_{\pi^j})\big)^2\mathcal{W}\big(b^{(\delta_t^i-2)}(\cdot|x_t^i)\,\big\|\,b_{\phi,\psi}^{(\delta_t^i-2)}(\cdot|x_t^i)\big)\\
\le&\cdots\\
\le&(\sum_{i\in\mathcal{N}}\epsilon_{\mathcal{P}}^i+L_{\mathcal{P}}\sum_{j\neq i}\epsilon_\pi^j)\sum_{k=0}^{\delta_t^i-1}\big(L_{\mathcal{P}}(1+\sum_{j\neq i}L_{\pi^j})\big)^k\\
&+\big(L_{\mathcal{P}}(1+\sum_{j\neq i}L_{\pi^j})\big)^{\delta_t^i}\mathcal{W}\big(b^{(0)}(\cdot|x_t^i)\,\big\|\,b_{\phi,\psi}^{(0)}(\cdot|x_t^i)\big)\\
=&(\sum_{i\in\mathcal{N}}\epsilon_{\mathcal{P}}^i+L_{\mathcal{P}}\sum_{j\neq i}\epsilon_\pi^j)\sum_{k=0}^{\delta_t^i-1}\big(L_{\mathcal{P}}(1+\sum_{j\neq i}L_{\pi^j})\big)^k\\
=&(\sum_{i\in\mathcal{N}}\epsilon_{\mathcal{P}}^i+L_{\mathcal{P}}\sum_{j\neq i}\epsilon_\pi^j)\frac{1-\big(L_{\mathcal{P}}(1+\sum_{j\neq i}L_{\pi^j})\big)^{\delta_t^i}}{1-L_{\mathcal{P}}(1+\sum_{j\neq i}L_{\pi^j})}.
\end{aligned}$$

Replacing an observed coordinate by the same known value in both predictions sets its coordinate error to zero. Availability indicators could therefore exclude these coordinate contributions from the one-step error bound $\sum_{i\in\mathcal{N}}\epsilon_{\mathcal{P}}^i + L_{\mathcal{P}}\sum_{j\neq i}\epsilon_\pi^j$. □

Proposition 1 (Lipschitz Continuous Q_π^i and V_π^i). *For a fixed stationary joint policy π satisfying Definition 1, if $\gamma L_{\mathcal{P}}\big(1+\sum_{j\neq i} L_{\pi^j}\big)\big(1+L_{\pi^i}\big) < 1$, then Q_π^i and V_π^i are Lipschitz continuous under $d_{\mathcal{S}} + d_{\mathcal{A}^i}$ and $d_{\mathcal{S}}$, respectively, with valid constants*

$$L_{Q^i} = \frac{L_{\mathcal{R}^i}(1+\sum_{j\neq i} L_{\pi^j})}{1-\gamma L_{\mathcal{P}}(1+\sum_{j\neq i} L_{\pi^j})(1+L_{\pi^i})}, \qquad L_{V^i} = L_{Q^i}(1+L_{\pi^i}).$$

Proof. Fix the stationary product policy π. Compactness and the Lipschitz reward assumption imply that rewards are bounded. Define the finite-horizon values by $V_{\pi,0}^i = 0$ and, for $n \geq 0$,

$$\begin{aligned} Q_{\pi,n+1}^i(s,a^i) &:= \mathbb{E}_{a^{-i}\sim\pi^{-i}(\cdot|s)}\left[\mathcal{R}^i(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V_{\pi,n}^i(s')]\right], \\ V_{\pi,n+1}^i(s) &:= \mathbb{E}_{a^i\sim\pi^i(\cdot|s)}[Q_{\pi,n+1}^i(s,a^i)]. \end{aligned}$$

The constants stated in the proposition are finite under its strict inequality and satisfy

$$L_{Q^i} = \left(1+\sum_{j\neq i} L_{\pi^j}\right)\left(L_{\mathcal{R}^i} + \gamma L_{\mathcal{P}} L_{V^i}\right), \qquad L_{V^i} = L_{Q^i}(1+L_{\pi^i}).$$

We show by induction that these constants bound the Lipschitz constants of all finite-horizon values. The base case $V_{\pi,0}^i = 0$ is immediate. Assume $V_{\pi,n}^i$ is L_{V^i}-Lipschitz.

Given (s_1, a_1^i) and (s_2, a_2^i), couple agents' actions by the product of their optimal couplings, and write $a_1 = (a_1^i, a_1^{-i})$ and $a_2 = (a_2^i, a_2^{-i})$. Under this coupling,

$$\begin{aligned} &\mathbb{E}[d_{\mathcal{S}}(s_1,s_2) + d_{\mathcal{A}}(a_1,a_2)] \\ &\quad = d_{\mathcal{S}}(s_1,s_2) + d_{\mathcal{A}^i}(a_1^i,a_2^i) + \sum_{j\neq i} \mathcal{W}\big(\pi^j(\cdot|s_1)\|\pi^j(\cdot|s_2)\big) \\ &\quad \leq \left(1+\sum_{j\neq i} L_{\pi^j}\right)\big(d_{\mathcal{S}}(s_1,s_2) + d_{\mathcal{A}^i}(a_1^i,a_2^i)\big). \end{aligned}$$

Using the induction hypothesis and Definition 1,

$$\begin{aligned} &|Q_{\pi,n+1}^i(s_1,a_1^i) - Q_{\pi,n+1}^i(s_2,a_2^i)| \\ &\quad \leq \mathbb{E}[|\mathcal{R}^i(s_1,a_1) - \mathcal{R}^i(s_2,a_2)|] + \gamma L_{V^i}\mathbb{E}[\mathcal{W}(\mathcal{P}(\cdot|s_1,a_1)\|\mathcal{P}(\cdot|s_2,a_2))] \\ &\quad \leq (L_{\mathcal{R}^i} + \gamma L_{\mathcal{P}} L_{V^i})\mathbb{E}[d_{\mathcal{S}}(s_1,s_2) + d_{\mathcal{A}}(a_1,a_2)] \\ &\quad \leq L_{Q^i}\left(d_{\mathcal{S}}(s_1,s_2) + d_{\mathcal{A}^i}(a_1^i,a_2^i)\right). \end{aligned}$$

Consequently,

$$\begin{aligned} &|V_{\pi,n+1}^i(s_1) - V_{\pi,n+1}^i(s_2)| \\ &\quad \leq \mathbb{E}_{a^i\sim\pi^i(\cdot|s_1)}[|Q_{\pi,n+1}^i(s_1,a^i) - Q_{\pi,n+1}^i(s_2,a^i)|] \\ &\qquad + \left|\mathbb{E}_{a^i\sim\pi^i(\cdot|s_1)}[Q_{\pi,n+1}^i(s_2,a^i)] - \mathbb{E}_{a^i\sim\pi^i(\cdot|s_2)}[Q_{\pi,n+1}^i(s_2,a^i)]\right| \\ &\quad \leq L_{Q^i} d_{\mathcal{S}}(s_1,s_2) + L_{Q^i}\mathcal{W}(\pi^i(\cdot|s_1)\|\pi^i(\cdot|s_2)) \\ &\quad \leq L_{Q^i}(1+L_{\pi^i}) d_{\mathcal{S}}(s_1,s_2) = L_{V^i} d_{\mathcal{S}}(s_1,s_2). \end{aligned}$$

This closes the induction. Bounded rewards and $\gamma < 1$ imply uniform convergence of the truncated returns to Q_π^i and V_π^i. Passing to the limit preserves the same Lipschitz inequalities. Thus the infinite-horizon functions admit the stated constants

$$L_{Q^i} = \frac{L_{\mathcal{R}^i}(1+\sum_{j\neq i} L_{\pi^j})}{1-\gamma L_{\mathcal{P}}(1+\sum_{j\neq i} L_{\pi^j})(1+L_{\pi^i})}, \qquad L_{V^i} = L_{Q^i}(1+L_{\pi^i}).$$

These are valid upper bounds, not necessarily the smallest Lipschitz constants. □

Corollary 1 (Joint Policy Difference). *Given policy π^i and x_t^i, we have*

$$\mathcal{W}\big(\pi^i(\cdot|b(\cdot|x_t^i))\big\|\ \pi^i(\cdot|b_{\phi,\psi}(\cdot|x_t^i))\big)\ \le\ L_{\pi^i}\epsilon_b^i.$$

For $\pi_{\phi,\psi} = (\pi^i_{\phi,\psi})_{i\in\mathcal{N}}$ and $\pi_b = (\pi_b^i)_{i\in\mathcal{N}}$, whose agents respectively act on $b_{\phi,\psi}$ and b, we have

$$\mathcal{W}\big(\pi_b(\cdot|x_t)\big\|\ \pi_{\phi,\psi}(\cdot|x_t)\big)\ \le\ \sum_{i\in\mathcal{N}} L_{\pi^i}\epsilon_b^i.$$

Proof. Fix x_t^i and couple $s_1 \sim b(\cdot|x_t^i)$ and $s_2 \sim b_{\phi,\psi}(\cdot|x_t^i)$ optimally. Conditional on (s_1, s_2), couple actions drawn from $\pi^i(\cdot|s_1)$ and $\pi^i(\cdot|s_2)$ optimally. This induces a coupling of the two belief-averaged action distributions, so

$$\begin{aligned}
&\mathcal{W}\big(\pi^i(\cdot|b(\cdot|x_t^i))\big\|\ \pi^i(\cdot|b_{\phi,\psi}(\cdot|x_t^i))\big)\\
\le&\mathbb{E}\left[\mathcal{W}\big(\pi^i(\cdot|s_1)\|\pi^i(\cdot|s_2)\big)\right]\\
\le&L_{\pi^i}\mathbb{E}\left[d_{\mathcal{S}}(s_1,s_2)\right]\\
=&L_{\pi^i}\,\mathcal{W}\big(b(\cdot|x_t^i)\big\|b_{\phi,\psi}(\cdot|x_t^i)\big)\\
\le&L_{\pi^i}\epsilon_b^i
\end{aligned}$$

For product action distributions under the additive action metric, the Wasserstein distance equals the sum of the marginal distances: a product of optimal marginal couplings attains this cost, and every joint coupling has at least this cost. Hence

$$\begin{aligned}
&\mathcal{W}\big(\pi_b(\cdot|x_t)\big\|\ \pi_{\phi,\psi}(\cdot|x_t)\big)\\
=&\sum_{i\in\mathcal{N}}\mathcal{W}\big(\pi^i(\cdot|b(\cdot|x_t^i))\big\|\ \pi^i(\cdot|b_{\phi,\psi}(\cdot|x_t^i))\big)\\
\le&\sum_{i\in\mathcal{N}} L_{\pi^i}\epsilon_b^i
\end{aligned}$$

□

Corollary 2 (Joint Action Value Difference). *Under Assumption 1, for fixed x_t, s_t and any joint action $a_t \in \mathcal{A}$, the outputs in Equation* (3) *satisfy*

$$\left|Q_b^{\text{total}}(s_t, a_t) - Q_{b_{\phi,\psi}}^{\text{total}}(s_t, a_t)\right| \leq L_f \sum_{i \in \mathcal{N}} L_{Q^i} \epsilon_b^i.$$

Proof. Keep the utility and mixer parameters, s_t, and a_t fixed. For each agent, couple its two belief distributions optimally. The state-Lipschitz utility condition gives

$$\left|\mathbb{E}_{b(\cdot|x_t^i)}[Q^i(s, a_t^i)] - \mathbb{E}_{b_{\phi,\psi}(\cdot|x_t^i)}[Q^i(s, a_t^i)]\right| \leq L_{Q^i} \mathcal{W}\big(b(\cdot|x_t^i) \| b_{\phi,\psi}(\cdot|x_t^i)\big).$$

Applying the mixer bound at the same state input yields

$$\begin{aligned}
&\left|Q_b^{\text{total}}(s_t, a_t) - Q_{b_{\phi,\psi}}^{\text{total}}(s_t, a_t)\right| \\
\leq & L_f \sum_{i \in \mathcal{N}} \left|\mathbb{E}_{s \sim b(\cdot|x_t^i)}\big[Q^i(s, a_t^i)\big] - \mathbb{E}_{s \sim b_{\phi,\psi}(\cdot|x_t^i)}\big[Q^i(s, a_t^i)\big]\right| \\
\leq & L_f \sum_{i \in \mathcal{N}} L_{Q^i} \mathcal{W}\big(b(\cdot|x_t^i) \| b_{\phi,\psi}(\cdot|x_t^i)\big) \\
\leq & L_f \sum_{i \in \mathcal{N}} L_{Q^i} \epsilon_b^i,
\end{aligned}$$

□

B IMPLEMENTATION DETAILS

B.1 EXPERIMENTAL PROTOCOL

Training and evaluation. All methods use QMIX with the 10M-step budget and 10 seeds in Section 6.1. Each sampled batch first updates the belief predictor, then the utilities using detached reconstructions and the curriculum of Fu et al. (2025). We evaluate each final checkpoint over 100 episodes with greedy action selection and report the mean and standard deviation across seeds of the per-run performance metrics.

Delay protocol. Following Fu et al. (2025), random delays samples nominal delays uniformly from $\{0, \ldots, 2d\}$. Components are retrieved from history with nondecreasing source timestamps, so realized staleness also depends on previously received observations. Self-state remains current while other agents and environmental entities are delayed.

Offline belief evaluation. We collect 1000 MPE episodes during delay-free QMIX training and split whole episodes into 900 training and 100 test episodes. All beliefs share the split and are trained separately for each delay and team size.

B.2 BEHAVIOUR SUPERVISION AND FEEDBACK

The cross-entropy targets in Equation (5) are recomputed during belief training. For each agent j and historical timestep k in the delay window, the policy network receives that agents' delay-free local observations at k. The resulting Q-values define an ε-greedy target distribution, rather than a softmax distribution over Q-values. These targets use delay-free local observations and need not coincide with the action distributions used during delayed execution; the Lipschitz reference-policy assumptions in the analysis do not automatically hold for ε-greedy policies. The behaviour head outputs a full action distribution via softmax. This predicted distribution is concatenated with token embeddings as feedback, without argmax or sampling. Reconstruction gradients pass through feedback to the behaviour head and shared encoder, as reflected by $\mathcal{L}_{\mathrm{rec}}(\psi)$. The shared encoder parameters are included in ψ, and ϕ denotes the state-head parameters. Only the subsequent QMIX utility update uses detached reconstructions.

Feedback ablations. Oracle feedback replaces predicted distributions with one-hot encodings of agents' actual actions during the delay window, using privileged information unavailable at execution. The w/o feedback variant retains the behaviour head and its supervision but removes feedback to the state head.

B.3 NETWORK CONFIGURATION AND OPTIMIZATION

BADA uses a shared transformer encoder with behaviour and state heads, trained with cross-entropy and MSE, respectively. Architecture, Adam settings, and default loss weights are listed in Table 2.

B.4 BASELINE INTEGRATION

All predictors supply reconstructed states to QMIX; the Delay-aware baseline uses delayed inputs and the delay-free reference omits the delay filter. We adapt DFBT (Wu et al., 2025), DUPO (Tu et al., 2026), and DDC (Mednikov & Gal, 2026) to the per-link delays of Section 3 as follows.

DFBT. Given BADA's information state, DFBT uses one token per timestep to forecast the delay window in a single pass. Its current-state prediction feeds the utilities.

DUPO. DUPO's denoiser receives BADA's information state; its sample mean feeds the utilities. QMIX replaces the original single-agent policy optimizer.

DDC. A per-agent Kalman filter uses learned GRU dynamics in place of linear transitions and is trained jointly with QMIX, rather than attached to a frozen policy.

Common comparison protocol. The adaptations share BADA's observations, seeds, budget, mixer, curriculum, and training schedule. Each setting uses a common delay-free QMIX reference.

B.5 HYPER-PARAMETERS

We provide the hyper-parameters of BADA in Table 2.

Table 2: Hyper-parameters table of BADA.

Hyper-parameter	Value
Hidden Dim	64
Attention Heads Num	4
Encoder Blocks Num	2
Feed-Forward Dim	128
Behaviour Loss Weight λ_{beh}	1
State Loss Weight λ_{rec}	3
Optimizer	Adam
Learning Rate	3e-3
Gradient Clipping	10
Environment Steps	1e7

C ADDITIONAL EXPERIMENTAL RESULTS

C.1 CLASSIC SMAX MAPS

On 2s3z, 3s5z, and 6h_vs_8z, several methods approach unit win rate, leaving little room for improvement (Table 3). BADA's gains are larger on 3s_vs_5z with constant delay 8 (0.88 versus RDC-TF's 0.75) and 27m_vs_30m with random delay 8 (0.97 versus DFBT and DDC's 0.79). It trails RDC-TF on 10m_vs_11m with constant delay 8 (0.31 versus 0.35). On 10m_vs_11m, BADA achieves 0.49 under random delay 8 ± 8 versus 0.31 under constant delay 8.

Delay	Delay-aware	RDC-GRU	RDC-TF	Recursive Belief	DFBT	DUPO	DDC	BADA(ours)
2s3z delay-free: 0.99 ± 0.03								
constant 8	0.78 ± 0.20	0.93 ± 0.19	0.97 ± 0.03	0.91 ± 0.01	0.93 ± 0.01	0.95 ± 0.01	0.97 ± 0.02	$\mathbf{0.99 \pm 0.01}$
random 8 ± 8	0.94 ± 0.04	$\mathbf{0.99 \pm 0.01}$	0.97 ± 0.03	0.97 ± 0.05	0.98 ± 0.01	0.93 ± 0.01	0.97 ± 0.02	$\mathbf{0.99 \pm 0.01}$
3s5z delay-free: 0.98 ± 0.02								
constant 8	0.91 ± 0.08	0.93 ± 0.08	0.95 ± 0.04	0.93 ± 0.07	0.93 ± 0.01	0.96 ± 0.04	0.97 ± 0.06	$\mathbf{0.98 \pm 0.03}$
random 8 ± 8	0.94 ± 0.04	0.96 ± 0.05	0.95 ± 0.11	0.98 ± 0.02	0.98 ± 0.01	0.98 ± 0.01	$\mathbf{0.99 \pm 0.01}$	$\mathbf{0.99 \pm 0.01}$
10m_vs_11m delay-free: 0.63 ± 0.02								
constant 8	0.01 ± 0.01	0.25 ± 0.04	$\mathbf{0.35 \pm 0.05}$	0.03 ± 0.01	0.29 ± 0.14	0.03 ± 0.01	0.02 ± 0.01	0.31 ± 0.20
random 8 ± 8	0.03 ± 0.01	0.27 ± 0.06	0.29 ± 0.04	0.14 ± 0.05	0.37 ± 0.01	0.23 ± 0.02	0.22 ± 0.07	$\mathbf{0.49 \pm 0.10}$
3s_vs_5z delay-free: 0.99 ± 0.03								
constant 8	0.12 ± 0.08	0.44 ± 0.21	0.75 ± 0.18	0.24 ± 0.20	0.62 ± 0.15	0.68 ± 0.29	0.62 ± 0.15	$\mathbf{0.88 \pm 0.11}$
random 8 ± 8	0.16 ± 0.07	0.79 ± 0.18	0.94 ± 0.05	0.71 ± 0.18	0.88 ± 0.24	0.97 ± 0.03	0.88 ± 0.24	$\mathbf{0.99 \pm 0.01}$
6h_vs_8z delay-free: 1.00 ± 0.00								
constant 8	0.83 ± 0.10	0.98 ± 0.02	0.99 ± 0.01	0.88 ± 0.12	0.96 ± 0.04	0.97 ± 0.03	0.96 ± 0.02	$\mathbf{1.00 \pm 0.00}$
random 8 ± 8	0.98 ± 0.02	0.99 ± 0.00	$\mathbf{1.00 \pm 0.01}$	0.99 ± 0.01	0.99 ± 0.01	0.89 ± 0.04	0.87 ± 0.30	$\mathbf{1.00 \pm 0.00}$
27m_vs_30m delay-free: 0.85 ± 0.25								
constant 8	0.24 ± 0.11	0.62 ± 0.11	0.74 ± 0.08	0.16 ± 0.05	0.52 ± 0.07	0.63 ± 0.11	0.51 ± 0.07	$\mathbf{0.88 \pm 0.11}$
random 8 ± 8	0.31 ± 0.06	0.35 ± 0.12	0.44 ± 0.12	0.65 ± 0.05	0.79 ± 0.07	0.57 ± 0.15	0.79 ± 0.07	$\mathbf{0.97 \pm 0.02}$

Table 3: Performance on the classic SMAX maps with different delays.

C.2 HANABI AND COIN GAME

The adapted single-agent predictors remain competitive: DFBT ties BADA at 2.46 in Hanabi with constant delay 8 (Table 4). In Coin Game with random delay 4, BADA reaches 15.91, ahead of DUPO's 13.54 and DFBT's 12.51, but far below the delay-free 54.03 (Table 5). Its leading return of 4.38 under constant delay 4 likewise leaves a large gap. Thus, baseline gains do not imply full compensation for delay.

Delay	Delay-aware	RDC-GRU	RDC-TF	Recursive Belief	DFBT	DUPO	DDC	BADA(ours)
Hanabi delay-free: 6.73 ± 1.74								
constant 4	2.31 ± 0.59	2.35 ± 0.22	2.42 ± 0.12	2.39 ± 0.18	3.03 ± 0.17	2.48 ± 0.16	2.19 ± 0.13	$\mathbf{3.24 \pm 0.13}$
random 4 ± 4	3.34 ± 0.26	2.98 ± 1.00	3.30 ± 0.42	3.52 ± 0.25	3.31 ± 0.19	3.22 ± 0.23	3.12 ± 0.20	$\mathbf{3.66 \pm 0.42}$
constant 8	1.54 ± 0.91	1.04 ± 0.63	1.25 ± 0.68	0.94 ± 0.65	$\mathbf{2.46 \pm 0.60}$	0.85 ± 0.51	0.92 ± 0.35	$\mathbf{2.46 \pm 0.60}$
random 8 ± 8	2.84 ± 0.75	2.82 ± 0.29	2.87 ± 0.39	3.32 ± 0.21	3.00 ± 0.12	2.75 ± 0.30	3.12 ± 0.23	$\mathbf{3.51 \pm 0.31}$

Table 4: Performance on Hanabi with different delays.

Delay	Delay-aware	RDC-GRU	RDC-TF	Recursive Belief	DFBT	DUPO	DDC	BADA(ours)
coin_game delay-free: 54.03 ± 0.11								
constant 4	0.02 ± 0.08	1.62 ± 0.57	1.77 ± 0.43	0.82 ± 0.16	2.26 ± 0.75	1.69 ± 0.69	0.62 ± 0.14	$\mathbf{4.38 \pm 0.78}$
random 4 ± 4	1.91 ± 2.42	12.14 ± 0.23	12.48 ± 0.24	7.94 ± 0.48	12.51 ± 0.30	13.54 ± 0.34	6.74 ± 0.42	$\mathbf{15.91 \pm 0.14}$

Table 5: Performance on Coin Game with different delays.