ON THE ROLE OF COMMUNICATION IN OFFLINE MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper presents CODA, a framework for learning inter-agent communication from offline multi-agent datasets collected without communication. CODA jointly learns action and communication policies from fixed offline data, without message supervision or additional environment interaction, and deploys the learned communication after training. We analyze the role of communication in offline multi-agent RL, where communication can provide additional information about dataset-supported actions but can also increase the difficulty of learning a message-conditioned actor. Our analysis formalizes this trade-off and provides a condition under which communication tightens an overestimation bound. Experiments across several multi-agent tasks show that CODA can substantially improve offline policy learning, with gains that depend on the underlying offline MARL algorithm and dataset. Further analysis shows that communication often reduces policy–data action deviation and, in several settings, empirical critic overestimation, consistent with the stabilization perspective suggested by the theory. CODA also remains effective under delayed communication in several tasks.

1 INTRODUCTION

A key to advanced multi-agent systems, coordination is defined as the effective management of dependencies among agents (Malone & Crowston, 1990; 1994; Espinosa et al., 2004). One way to enhance coordination is introducing effective communication, which induces explicit dependencies among agents. Accordingly, learning communication has been actively studied in multi-agent RL and shown to improve performance in environments requiring coordination (Foerster et al., 2016; Sukhbaatar et al., 2016; Havrylov & Titov, 2017; Du et al., 2021; Sheng et al., 2022). Despite its effectiveness, much of the work on communication learning considers the online setting, where agents can interact with the environment to generate samples that include communication.

Offline RL, which trains policies from a fixed dataset without further environment interaction, offers a practical alternative when collecting additional interaction data is costly or impractical (Kumar et al., 2020; Fujimoto & Gu, 2021; Kostrikov et al., 2022). A primary challenge is extrapolation error when policies select actions poorly supported by the dataset, which can lead to value overestimation (Kumar et al., 2020). In offline multi-agent RL, this challenge is compounded by the large joint action space, motivating behavior-constrained and conservative approaches (Pan et al., 2022; Wang et al., 2023; Shao et al., 2023; Kim & Sycara, 2026).

Given the recent progress in offline multi-agent RL, one might naturally ask: *Can we learn effective communication from offline datasets?* We focus on datasets generated by non-communicating agents and ask whether useful communication can be learned without recorded messages. A priori, it is unclear what role communication plays in the offline setting: conditioning the actor on messages enlarges the policy class and could amplify extrapolation error, while messages may reveal latent context that local observations alone cannot distinguish. Our analysis makes this trade-off explicit. First, we show that adding messages cannot increase residual uncertainty about a latent-context-dependent dataset action center. We then derive a sufficient condition under which the resulting information gain outweighs the additional actor-learning error and tightens an overestimation bound. Building on this insight, we propose CODA, which jointly learns action and communication policies from offline data without message supervision or additional environment interaction. As illustrated in Fig. 1, the learned policies are deployed in environments where agents can communicate.

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.

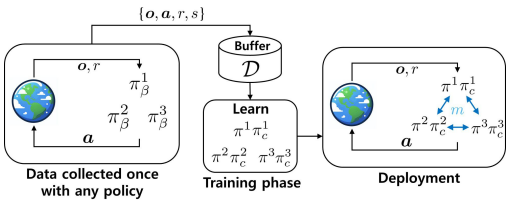


Figure 1: Learning both the multi-agent policy and communication policy from a given offline dataset generated by non-communicating agents. After training, the learned policies π^i and π^i_c are deployed in the test phase. m denotes the messages communicated between agents.

We evaluate CODA with two offline MARL algorithms, B3C (Kim & Sycara, 2026) and OMIGA (Wang et al., 2023), across multi-agent MuJoCo and predator–prey tasks with varying partial observability and dataset quality. With B3C, CODA shows broad performance gains, whereas improvements with OMIGA are less consistent across settings, indicating that the benefit of communication depends on the underlying offline learning algorithm. Further analyses show that communication often keeps learned actions closer to the dataset actions and can reduce empirical overestimation, consistent with the stabilization perspective suggested by the theory. Comparisons with parameter-matched and direct observation-sharing baselines show that the gains are not explained by added capacity alone, supporting that CODA learns useful inter-agent communication from the offline data. CODA also remains effective under delayed communication in several settings. Our contributions are as follows:

- We introduce the problem of learning communication from offline datasets that lack communication data, and provide an end-to-end framework for learning both action and communication policies.
- We provide an analysis identifying when communication tightens an overestimation bound in offline multi-agent RL, framing it as a trade-off between information gain and an actor-class cost.
- We evaluate CODA with B3C and OMIGA across multiple environments and datasets, and analyze its offline learning behavior, attribution of the gains, and delayed communication.

2 Background and Related Works

2.1 Multi-Agent RL

We consider fully cooperative multi-agent tasks modeled as a decentralized partially observable Markov decision process (Dec-POMDP) with N agents (Oliehoek, 2012). At each time step t, the environment maintains a global state s_t, while each agent i receives a local observation o^i_t and selects an action a^i_t according to its policy π^i. The joint action $\boldsymbol{a}_t = (a^1_t, \cdots, a^N_t)$ determines the next state s_{t+1}, the subsequent observations, and a shared reward r_t. The objective is to learn a joint policy $\boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{\tau}) = \prod_{i=1}^N \pi^i(a^i|\tau^i)$ that maximizes the cumulative team reward $\mathbb{E}\left[\sum_{l=0}^\infty \gamma^l r_l\right]$, where τ^i is the local action-observation history of agent i.

2.2 Learning Communication in Multi-Agent RL

To enhance coordination under decentralized execution, multi-agent RL has explored learning communication, where agents exchange messages through a differentiable channel and incorporate them into decision-making. This allows agents to share partial observations (Foerster et al., 2016; Das et al., 2019) and intentions (Kim et al., 2021; Fang et al., 2023), leading to improved performance. A common approach is to train a communication policy via the channel to optimize other agents' policy and critic losses. For example, DIAL optimizes communication based on gradients of other agents' Q-function losses (Foerster et al., 2016), while intention sharing uses imagined trajectories as messages, extending beyond observation sharing (Kim et al., 2021). Other works study when (Jiang & Lu, 2018; Kim et al., 2019a) and whom to communicate with (Das et al., 2019), aiming for efficiency with low complexity (Kim et al., 2019b; Karten et al., 2023). These studies motivate examining communication learning in offline settings.

2.3 Offline Multi-Agent RL

Offline RL aims to learn a policy that maximizes the expected return from a fixed dataset $\mathcal{D}$ of trajectories generated by arbitrary behavior policies, without further environment interaction. Existing approaches include behavior-constrained methods (Fujimoto & Gu, 2021; Xu et al., 2023), return-conditioned supervised learning (Chen et al., 2021; Kim et al., 2024a), value-based methods (Kumar

et al., 2020; Kostrikov et al., 2022), and hybrid approaches (Kim et al., 2024b). We focus on behavior-constrained methods, which mitigate overestimation by regularizing the learned policy toward the behavior policy. Two representative single-agent instances apply the divergence between the learning and behavior policies as either an additional reward (Xu et al., 2023) or a behavior cloning (BC) loss (Fujimoto & Gu, 2021). In multi-agent settings, overestimation becomes more severe as the joint action space grows exponentially with the number of agents (Yang et al., 2021), and value factorization is often needed to assign credit for shared rewards. To tackle these challenges, numerous studies have been conducted (Jiang & Lu, 2023; Tseng et al., 2022; Meng et al., 2023; Zhu et al., 2024). We build on two multi-agent extensions of behavior-constrained methods.

We primarily build on FACMAC+B3C (Kim & Sycara, 2026), referred to as B3C for brevity. B3C combines FACMAC's factored centralized critic (Peng et al., 2021) with an action-space BC penalty and clips the bootstrapped Q-value using the maximum dataset return to limit critic divergence. Writing $\tilde{\boldsymbol{a}}_\pi$ for the sampled actor action, the joint policy is updated by

$$\max_{\boldsymbol{\pi}} \mathbb{E}_{\mathcal{D}}\left[c_\alpha \widehat{Q}_{jt}(s, \boldsymbol{o}, \tilde{\boldsymbol{a}}_\pi) - \beta\, \ell_{\mathrm{BC}}(\tilde{\boldsymbol{a}}_\pi, \boldsymbol{a})\right], \tag{1}$$

where c_α is a detached normalization coefficient controlled by α. Larger α places more weight on the RL objective relative to BC; we fix $\beta = 1$. This objective makes explicit the offline trade-off between maximizing the learned value and remaining close to dataset-supported actions. The exact normalization and implementation details are provided in Appendix B.

We also use OMIGA (Wang et al., 2023) to examine a different policy-learning objective. OMIGA factorizes joint Q- and V-functions, incorporates reverse-KL regularization, and extracts decentralized actors by advantage-weighted likelihood of logged actions, rather than B3C's direct critic-based actor update. Its full actor objective and implementation details are given in Appendix B.3.

3 METHODOLOGY

3.1 SETTING

We introduce a new problem in offline multi-agent RL with communication. Specifically, given a multi-agent dataset that does not include any communication among agents, we aim to train both decentralized policies and communication policies that can be deployed in environments where communication is available, in order to maximize the expected return.

Dataset. The dataset $\mathcal{D}$ is assumed to consist of D synchronized episodes, $\mathcal{D} = \{\boldsymbol{s}_d, \boldsymbol{o}_d, \boldsymbol{a}_d, \boldsymbol{r}_d\}_{d=1}^{D}$. Here, $\boldsymbol{s}_d$, $\boldsymbol{o}_d$, $\boldsymbol{a}_d$, and $\boldsymbol{r}_d$ are the sequences of states, joint observations, joint actions, and rewards in the d-th episode. For example, $\boldsymbol{o}_d = (\boldsymbol{o}_{d,t=1}, \cdots, \boldsymbol{o}_{d,t=T})$, where $\boldsymbol{o}_{d,t} = (o_{d,t}^{i=1}, \cdots, o_{d,t}^{i=N})$, and $o_{d,t}^i$ is Agent i's observation at time-step t of d-th episode. The data contain no messages, but include the simultaneous local observations needed to compute messages during offline training. Note that the dataset can be generated by arbitrary non-communicating policies. In this paper, we use datasets generated by existing online multi-agent algorithms.

Multi-agent RL with Communication. Given the dataset, we train multiple agents, each with a policy and a communication policy, to maximize the cumulative sum of rewards, and then deploy them in environments where multiple agents can communicate with each other. We follow a scenario of communication built on top of the Dec-POMDP described in Sec. 2.1, which has been widely considered in the literature (Sukhbaatar et al., 2016; Singh et al., 2019). That is, at time step t, Agent i observes an observation o_t^i and communicates messages, $m_t = (m_t^1, \cdots, m_t^N)$, where m_t^j is the message from Agent j. Here, Agent i has an action policy $\pi^i(a_t^i | o_t^i, m_t)$ and a communication policy $\pi_c^i(m_t^i | o_t^i)$. The communication policy is a deterministic encoder in our implementation. Messages affect the environment only through the selected joint action; the transition and reward functions are otherwise unchanged. No new environment trajectories are collected for policy optimization; environment rollouts are used for evaluation.

3.2 LEARNING COMMUNICATION FROM DATASETS

We now present a framework that learns to **C**ommunicate from **O**ffline **DA**tasets, named `CODA`. `CODA` enables multiple agents to improve coordination by learning effective communication from a given dataset without collecting new environment interactions.

The key idea is to augment each agent with a communication policy $\pi_c^i(m^i \mid o^i)$ alongside its action policy $\pi^i(a^i \mid o^i, m)$, and to train both jointly by maximizing the actor objective of a behavior-constrained offline multi-agent RL algorithm. Since the dataset contains no messages, the communication policy receives no direct supervision; instead, it is shaped indirectly through gradient flow from all agents' actor objectives, while the underlying behavior-related objective discourages excessive deviation from the logged behavior. We incorporate CODA into two existing offline multi-agent RL methods, OMIGA and FACMAC+B3C, yielding OMIGA+CODA and FACMAC+B3C+CODA, respectively. The next section analyzes when this design is expected to help, and Sec. 3.4 details the architecture and training objectives.

3.3 Theoretical Analysis

We provide a simple analysis explaining when communication can tighten an overestimation bound in offline multi-agent RL. Under partial observability, messages from other agents can act as proxy information about the latent context that determines which actions are well supported by the offline dataset. At the same time, conditioning the actor on messages enlarges the policy space, which may increase offline learning error. The analysis below formalizes this trade-off; full statements of assumptions and proofs are provided in Appendix A.

Setup. We adopt the Dec-POMDP setup of Sec. 2.1, augmented with messages exchanged among agents, and use uppercase letters for the corresponding random variables. From the perspective of agent i, let O^i be its local observation, M^{-i} the messages received from other agents, and U^i a latent context that determines the dataset action center (e.g., the global state or the trajectory-quality mode of an episode). The actor information is $Z_0^i = O^i$ without communication and $Z_c^i = (O^i, M^{-i})$ with communication. We compare these information sets under one common reference law and a fixed learned communication map. We define $\bar{a}_D^i(u) = \mathbb{E}_D[A^i \mid U^i = u]$ as an oracle support-anchor, and the residual support uncertainty as

$$\mathcal{U}(Z^i) = \mathbb{E}\big[\mathrm{Var}\big(\bar{a}_D^i(U^i) \mid Z^i\big)\big], \tag{2}$$

where vector variance is the trace of conditional covariance. The anchor and actors have finite second moments under the common reference law. This center is an analytical device, not a full characterization of a potentially multimodal dataset action support.

Assumptions (informal). We use two explicit modeling assumptions (Appendix A). First, the pointwise positive critic error is bounded by the actor's distance from $\bar{a}_D^i(U^i)$, with common multiplier $\kappa > 0$ and additive error $\epsilon_Q \geq 0$. Second, the learned deterministic actor approximately minimizes squared anchor deviation given its information, up to a class-dependent error $\epsilon_\pi(Z^i)$. This is a one-agent comparison with the other agents' action rules held fixed. Conditioning on messages may increase the learned actor error, although a larger class does not inevitably do so. Behavior regularization motivates controlling this error; we do not assume that either backbone guarantees a small difference between the two error terms.

Proposition 1 (Communication reduces residual support uncertainty)**.**

$$\mathcal{U}(Z_0^i) - \mathcal{U}(Z_c^i) \geq 0. \tag{3}$$

The gap is zero when M^{-i} is conditionally independent of $\bar{a}_D^i(U^i)$ given O^i.

The reduction quantifies how much messages refine the prediction of the dataset action center beyond the local observation. It is nonnegative, but need not be large. Let π_Z^i denote the learned deterministic actor used in the analysis, and define $\mathrm{Over}(\pi_Z^i) = \mathbb{E}[(\widehat{Q} - Q^\pi)_+]$ at its policy-induced action.

Theorem 1 (Information-induced support stabilization)**.** *Under Assumptions 1 and 2,*

$$\mathrm{Over}(\pi_Z^i) \leq \kappa\sqrt{\mathcal{U}(Z^i) + \epsilon_\pi(Z^i)} + \epsilon_Q. \tag{4}$$

The bound separates residual support uncertainty $\mathcal{U}(Z^i)$, which communication can reduce, from the learned actor's excess error $\epsilon_\pi(Z^i)$, which communication may increase. Comparing the displayed bounds gives the following condition.

Corollary 1 (When communication tightens the bound)**.** *With the same κ and ϵ_Q in both comparisons, communication yields a no larger overestimation bound whenever*

$$\underbrace{\mathcal{U}(Z_0^i) - \mathcal{U}(Z_c^i)}_{\textit{information gain}} \geq \underbrace{\epsilon_\pi(Z_c^i) - \epsilon_\pi(Z_0^i)}_{\textit{actor-class cost}}. \tag{5}$$

The condition has an intuitive interpretation. The bound improves when the information provided about dataset-supported action centers outweighs the additional error of the learned actor. It compares upper bounds, rather than establishing an ordering of actual critic errors or returns.

Implications. The analysis identifies an opportunity for communication when local observations leave substantial action-center uncertainty and other agents' messages resolve it. Partial observability alone does not ensure a large gain, and redundant messages may give none. To investigate this explanation, we separately examine message-based action predictability, policy–data action deviation, and positive critic–return gaps. These empirical quantities are related diagnostics, rather than direct observations of the oracle terms in the bound.

3.4 Communication Policy and Training

Each agent has an action policy $\pi^i(a^i \mid o^i, m)$ and a communication policy $\pi_c^i(m^i \mid o^i)$. In our no-delay model, π_c^i generates an outgoing message from the agent's local observation, without using received messages. A fixed agent identity is also included when parameters are shared. Following prior work on attentional communication (Jiang & Lu, 2018; Niu et al., 2021), multi-head attention (Vaswani et al., 2017) combines the exchanged messages $m = (m^1, \ldots, m^N)$ into an agent-specific representation c^i for action selection. The action policy uses its local observation and c^i; the notation $\pi^i(a^i \mid o^i, m)$ includes this aggregation. Thus local message generation and message aggregation are distinct steps. The full architecture is provided in Appendix B. During each offline update, the communication policies generate messages from the synchronized local observations in the sampled data, so no recorded messages are required.

We jointly train the action policies, sender encoders, and attention module through the underlying offline actor objective. For B3C+CODA, writing $\tilde{\boldsymbol{a}}_{\pi,\pi_c}$ for actions generated using communication, Eq. (1) becomes

$$\max_{\boldsymbol{\pi},\boldsymbol{\pi}_c} \mathbb{E}_{\mathcal{D}}\left[c_\alpha \widehat{Q}_{jt}(s, \boldsymbol{o}, \tilde{\boldsymbol{a}}_{\pi,\pi_c}) - \beta \ell_{\mathrm{BC}}(\tilde{\boldsymbol{a}}_{\pi,\pi_c}, \boldsymbol{a})\right]. \tag{6}$$

Gradients pass through the receiving actors to the sender encoders, so each sender can learn from other agents' objectives even though its message is generated locally. For OMIGA+CODA, we likewise condition the weighted-likelihood actor on messages and jointly optimize the action and communication modules, as detailed in Appendix B.3.

4 Experiments

Motivated by the analysis in Sec. 3.3, we evaluate not only whether communication can be learned from message-free offline data, but also how it affects offline learning behavior, whether the gains can be explained by simpler alternatives, and whether they persist under communication delay.

(RQ1) Can useful communication be learned from datasets generated by non-communicating agents?
(RQ2) How does communication affect policy–data action deviation and critic–return gaps?
(RQ3) Are the gains explained by additional network capacity or direct observation sharing?
(RQ4) Can useful communication be learned when messages are delayed?

Sec. 4.2 addresses RQ1, while Secs. 4.3.1, 4.3.2, and 4.3.3 address RQ2–RQ4, respectively.

4.1 Experimental Setup

Environments and Offline Datasets. We evaluate our approach in three types of environments, namely fully observable (FO) multi-agent MuJoCo, partially observable multi-agent MuJoCo, and a three-agent partially observable predator–prey scenario introduced by Peng et al. (2021). The fully observable experiments consider 3-agent Hopper (Hop), 2-agent Ant (Ant), and 6-agent HalfCheetah (HC). The datasets originate from OMIGA (Wang et al., 2023) and were generated by HAPPO (Kuba et al., 2022). Each task uses expert (e), medium (m), medium-replay (mr), and medium-expert (me) data. These 12 comparisons are reported in the fully observable block of Table 1.

The partially observable (PO) MuJoCo experiments consider 6-agent HalfCheetah with $k = 0, 1$ and 5-agent Swimmer with $k = 0$, where k specifies the local observation neighborhood. Their datasets originate from FACMAC+B3C (Kim & Sycara, 2026) and were generated by ADER (Kim & Sung,

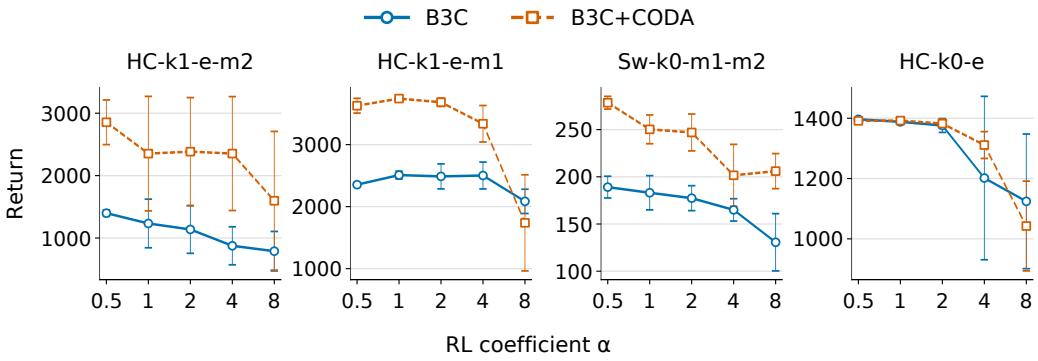


Figure 2: Return versus B3C's RL coefficient α for B3C and B3C+CODA. Points are five-seed mean returns over the final 20% of evaluations; error bars are SD. Increasing α increases the relative value-optimization weight while $\beta = 1$ in Eq. 6 is fixed. All 21 settings are shown in Appendix G.

2023). Each configuration uses expert (e), medium-1 (m1), medium-2 (m2), and their three pairwise mixtures, denoted by hyphenated labels. In predator–prey, agents observe prey within radius 0.5; we use e, m1, and e–m1 datasets. These give 21 partially observable task–dataset combinations. All datasets were collected without communication. The comparisons cover multiple observability conditions and datasets, rather than isolating observability as a single controlled factor.

Reported baselines. For context, Table 1 includes ICQ, OMAR, and CFCQL results reported in Wang et al. (2023); Kim & Sycara (2026), marked †. Our primary PO comparisons are between each CODA integration and its corresponding backbone under the same protocol.

Training and Evaluation. The PO performance grid uses $\alpha \in \{0.5, 1, 2, 4, 8\}$ for B3C and B3C+CODA, and $\alpha \in \{1, 10\}$ for OMIGA and OMIGA+CODA, with five seeds per cell. For each method and setting, α is selected using mean return over the final 20% of evaluations. The same runs are used for selection and reporting, so this grid is descriptive rather than a held-out tuning evaluation. Separate fixed-α blocks are used for attribution and mechanism analyses. The reporting rules for each block are provided in Appendix C.

4.2 Overall Performance

Table 1 summarizes the overall performance across all environments.

Partially observable environments. Our main evaluation compares each CODA integration with its own backbone in PO MuJoCo and predator–prey. B3C+CODA has a higher selected-α mean in 19/21 settings, with a median relative gain of 8.64%, including large gains on HC-k1-e-m2, Sw-k0-e-m2, and HC-k1-e-m1. In contrast, OMIGA+CODA improves 9/21 settings, with a median relative change of -0.10%. Thus, communication is broadly beneficial with B3C but its benefit depends on the offline learning backbone.

Fully observable environments. In the FO block of Table 1, B3C+CODA offers mostly small gains over B3C. OMIGA benefits in several settings, including Hopper medium-replay, where OMIGA+CODA outperforms the considered baselines and B3C also benefits from communication. Regressions occur as well, including B3C+CODA on Ant medium-replay. The FO and PO blocks use their respective experimental protocols. Taken together, the PO results show that communication can be learned from message-free data, while its performance benefit depends on the offline learning backbone.

Dependence on regularization. The complete PO grid also supports comparisons at a common α without selecting one value per method. Fig. 2 shows strong-gain, cross-environment, and low-gain examples. On HC-k1-e-m2, CODA retains a substantial advantage across the tested range. However, on HC-k1-e-m1 it underperforms B3C at $\alpha = 8$ despite large gains at smaller values. Performance advantages across several α values should therefore not be equated with uniformly lower hyperparameter sensitivity or universally smaller extrapolation error.

Table 1: Performance across 33 task–dataset combinations, reported as mean ± standard deviation (SD) over five seeds. Bold indicates the higher mean between each method and its CODA extension. Results marked † are from Wang et al. (2023); Kim & Sycara (2026).

Task–Dataset	ICQ†	OMAR†	CFCQL†	OMIGA	B3C	OMIGA+CODA	B3C+CODA
Fully observable multi-agent MuJoCo							
Hop-e	755.7	2.4	718.5	1244.9±424.8	**3619.7±1.6**	1031.5±672.4	3592.8±25.1
Hop-m	501.8	21.3	674.2	975.5±294.4	3242.7±129.1	1091.2±51.6	**3253.8±33.8**
Hop-mr	195.4	3.3	1380.2	840.5±249.1	736.8±469.4	**1622.0±441.8**	1327.9±269.0
Hop-me	355.4	1.4	383.0	1313.4±647.2	3328.0±369.8	1430.6±817.5	**3572.1±60.9**
Ant-e	2050.0	313.5	1756.1	2048.6±11.8	2162.8±46.0	2052.0±6.2	**2193.3±56.3**
Ant-m	1412.4	-1710.0	1159.6	1421.4±10.1	**1516.5±14.8**	1414.6±3.3	**1521.5±838.1**
Ant-mr	1016.7	-2014.2	1052.9	1085.8±67.9	**1259.8±302.4**	1022.3±54.8	954.0±78.1
Ant-me	1590.2	-2992.8	613.2	1291.8±869.0	**2077.6±194.2**	1894.7±44.1	**2080.1±154.1**
HC-e	2955.9	-206.7	4999.2	3148.2±172.5	**5403.5±169.7**	3674.8±441.9	**5447.2±453.3**
HC-m	2549.3	-265.7	4345.0	2684.2±43.8	**4756.6±56.3**	2544.0±85.8	**4707.6±191.6**
HC-mr	1922.4	-235.4	3655.3	2339.4±205.6	4602.6±150.2	2469.0±164.0	**4798.7±33.2**
HC-me	2834.0	-253.8	5030.9	2809.2±1021.1	5413.7±99.4	2622.5±77.6	**5517.5±45.1**
Partially observable multi-agent MuJoCo							
HC-k0-e	-	197.2	750.7	1389.4 ± 3.3	**1396.4 ± 4.3**	**1393.9 ± 5.4**	1392.2 ± 7.6
HC-k0-m1	-	189.6	443.7	1110.6 ± 4.9	1160.9 ± 22.4	**1112.4 ± 6.8**	**1187.7 ± 23.0**
HC-k0-m2	-	839.1	766.5	**833.7 ± 16.9**	1172.4 ± 47.2	832.5 ± 2.4	**1333.4 ± 46.9**
HC-k0-e-m1	-	136.4	542.6	**1265.5 ± 33.5**	1311.8 ± 23.0	1171.8 ± 25.2	**1384.6 ± 32.6**
HC-k0-e-m2	-	299.2	398.4	**1176.0 ± 11.8**	1223.6 ± 30.7	942.6 ± 22.1	**1312.6 ± 42.3**
HC-k0-m1-m2	-	709.1	1186.1	**1089.5 ± 20.3**	1150.7 ± 77.4	1064.0 ± 19.8	**1350.6 ± 20.6**
HC-k1-e	-	3232.6	3390.2	**3768.3 ± 3.4**	3772.6 ± 12.2	3766.2 ± 3.4	**3777.7 ± 27.5**
HC-k1-m1	-	2312.4	2356.0	1996.4 ± 8.1	2376.0 ± 47.4	**2002.8 ± 15.0**	**2597.4 ± 37.4**
HC-k1-m2	-	1108.9	1440.6	1241.1 ± 34.3	1924.8 ± 423.5	**1245.4 ± 12.0**	**2131.8 ± 481.4**
HC-k1-e-m1	-	2296.4	1949.6	**3546.4 ± 80.5**	2509.1 ± 63.4	2716.3 ± 282.0	**3742.6 ± 35.8**
HC-k1-e-m2	-	524.8	864.7	**1555.7 ± 334.5**	1398.8 ± 51.0	1543.7 ± 236.8	**2855.0 ± 358.0**
HC-k1-m1-m2	-	1410.6	1862.2	**1694.8 ± 29.4**	2088.5 ± 225.4	1551.3 ± 52.9	**2248.8 ± 410.3**
Sw-k0-e	-	395.3	403.3	**430.7 ± 0.2**	429.8 ± 0.8	429.9 ± 1.4	**430.1 ± 0.3**
Sw-k0-m1	-	268.4	277.1	291.6 ± 0.4	**290.4 ± 0.4**	**291.7 ± 0.7**	290.4 ± 0.4
Sw-k0-m2	-	97.9	130.0	**142.5 ± 1.3**	146.1 ± 1.9	142.3 ± 0.8	**148.0 ± 1.7**
Sw-k0-e-m1	-	216.5	292.1	276.6 ± 1.9	280.3 ± 7.3	**295.8 ± 6.1**	**371.1 ± 17.5**
Sw-k0-e-m2	-	113.8	138.6	**166.9 ± 26.2**	182.3 ± 29.4	158.4 ± 36.0	**364.5 ± 36.2**
Sw-k0-m1-m2	-	226.4	222.7	**210.1 ± 12.8**	189.1 ± 11.5	198.6 ± 12.5	**278.5 ± 6.7**
Partially observable predator–prey							
PP0.5-e	-	-	-	192.6 ± 2.4	184.9 ± 3.1	**193.3 ± 2.1**	**185.5 ± 3.0**
PP0.5-m1	-	-	-	40.0 ± 0.4	43.1 ± 1.5	**40.1 ± 0.9**	**46.8 ± 5.3**
PP0.5-e-m1	-	-	-	95.3 ± 16.5	104.0 ± 6.8	**140.3 ± 2.7**	**142.6 ± 23.4**

4.3 Analysis of Communication Gains

4.3.1 Impact of Communication on Offline Learning

To examine the support-stabilization perspective in Sec. 3.3, we compare policy–data action deviation and critic overestimation at a common α within each setting. Using a common α keeps the relative weighting between the RL objective and BC fixed, so differences are not simply explained by different regularization strengths. On fixed logged-data subsets, D_{act} is the mean per-agent squared distance between deterministic policy actions and logged actions, measuring how far the learned policy moves from the actions supported by the dataset. On evaluation rollouts, let $G_t = \sum_{k=t}^{T-1} \gamma^{k-t} r_k$ and $\delta_t = \widehat{Q}_{jt}(s_t, \boldsymbol{o}_t, \boldsymbol{a}_{\pi,t}) - G_t$. The positive gap $\mathbb{E}[\delta_+]$ measures how much the critic exceeds the realized return and serves as an empirical overestimation proxy. Together, these quantities pro-

Table 2: Return, policy–data action deviation, and empirical overestimation. Smaller D_{act} indicates actions closer to the dataset actions, while smaller $\mathbb{E}[\delta_+]$ indicates less empirical overestimation. Arrows denote B3C → B3C+CODA.

Setting	α	Return	Action deviation D_{act}	Overestimation $\mathbb{E}[\delta_+]$ (proxy)
HC-k0-e	1	1391.3 → 1392.3	0.00024 → 0.00024	12.26 → 13.23
HC-k0-m2	8	1172.4 → 1331.2	0.01034 → 0.01405	20.89 → 30.47
HC-k0-m1-m2	8	1115.7 → 1343.4	0.01583 → 0.01268	24.97 → 25.16
HC-k1-m2	8	1867.1 → 1960.2	0.01117 → 0.01032	74.29 → 28.89
HC-k1-e-m1	1	2530.4 → 3749.2	0.00350 → 0.00174	83.49 → 23.31
HC-k1-e-m2	0.5	1412.5 → 2757.7	0.00312 → 0.00123	83.73 → 82.98
Sw-k0-m1-m2	0.5	189.1 → 278.4	0.00595 → 0.00170	3.50 → 3.64

vide action- and value-level diagnostics related to the stabilization mechanism suggested by the theory. Appendix F details their relation to the oracle quantities, the evaluation conventions, and the signed and absolute gaps.

Table 2 shows that CODA reduces action deviation in five of the seven settings, alongside higher mean returns. The reductions are especially large on HC-k1-e-m1, HC-k1-e-m2, and Sw-k0-m1-m2. HC-k1-e-m1 and HC-k1-m2 also show substantial reductions in the overestimation proxy, from 83.49 to 23.31 and from 74.29 to 28.89, respectively, with greater variability in the latter. HC-k0-e has little change in either return or action deviation. Fig. 3 highlights these two cases together with HC-k1-e-m2, where action deviation decreases but the overestimation proxy changes little near the end of training. Results for all seven settings and their uncertainty estimates are provided in Appendix F.

These observations are consistent with the stabilization perspective motivated by Corollary 1. Messages can resolve ambiguity about dataset-supported action centers; when this information gain outweighs the added actor-learning error, the overestimation bound tightens. The action-deviation reductions support this perspective, while the critic response varies. HC-k1-e-m2 improves return and action fitting with little positive-gap change; HC-k0-m2 improves return despite increases in both diagnostics. Differences in dataset information and learned actors may change the balance between the two theoretical terms, which our measurements do not separately identify. Support stabilization is therefore one pathway to improved performance, not a necessary condition.

The OMIGA comparison provides a contrasting case: communication can substantially reduce policy–data action deviation without improving return, showing that closer action fitting alone is not sufficient for policy improvement. Detailed OMIGA diagnostics are provided in Appendix F. Overall, communication often reduces policy–data action deviation and can reduce empirical overestimation, but these effects depend on the setting and backbone.

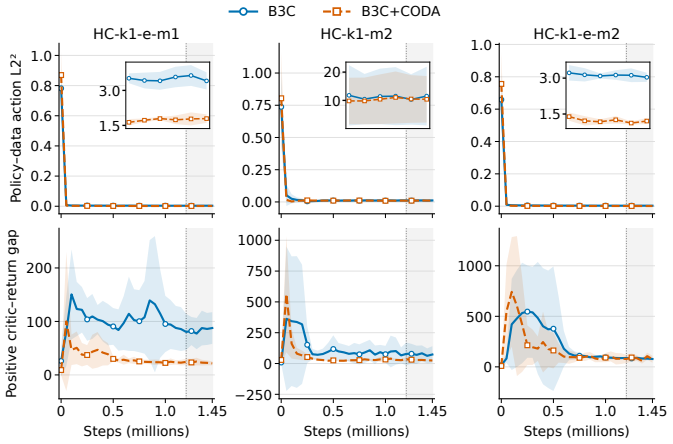


Figure 3: Policy–data action deviation and positive critic–return gap during training. Smaller action deviation (top) means the learned actions stay closer to actions in the dataset, while a smaller positive critic–return gap (bottom) indicates less empirical overestimation.

In addition, we conduct a separate post-training analysis by fitting an action-prediction probe to test whether other agents' messages contain information about the receiver's own action in the dataset. The learned messages substantially improve action prediction in several settings, as detailed in Appendix E.

4.3.2 Network Capacity and Observation Sharing

To test whether the gains come simply from a larger actor or from access to other agents' observations, we compare B3C and B3C+CODA with a parameter-matched no-communication variant and a direct observation-sharing baseline. *PM* matches the actor parameter count of B3C+CODA but blocks cross-agent information, while *RawObs* directly shares all agents' current local observations without

Table 3: Return comparison for learned communication, direct observation sharing, and no-communication baselines.

Setting	B3C	PM	RawObs	B3C+CODA
HC-k1-e-m1	2446.8±95.5	2465.3±43.9	3736.2±53.4	3727.9±79.5
HC-k1-e-m2	1154.1±370.0	948.6±319.4	2721.6±1172.2	2433.0±1290.0
HC-k1-m2	2007.2±290.4	1840.2±350.9	2251.7±143.9	2059.2±363.5
HC-k0-m1-m2	1081.0±41.5	1084.8±30.8	1219.5±19.0	1174.3±28.2
HC-k0-e	1394.4±10.0	1394.1±14.0	1383.7±42.9	1392.0±15.0
Sw-k0-e-m2	174.8±23.7	177.5±22.1	229.0±73.0	286.3±105.0
Sw-k0-m1-m2	181.5±7.0	178.6±12.2	273.0±17.5	278.4±12.8
PP0.5-e-m1	49.9±38.8	51.0±28.9	63.0±13.2	95.8±22.0

learned messages. All four methods use the same fixed α within each setting; details are in Appendix C.4.

Table 3 shows that PM remains close to B3C in most settings, indicating that the main gains are not explained by additional model capacity alone. Despite not directly sharing other agents' observations, B3C+CODA is often comparable to or better than RawObs, suggesting that the learned messages convey useful cross-agent information.

4.3.3 Learning with Delayed Communication

Communication delays can arise in practical multi-agent systems. We therefore evaluate whether CODA can still learn useful communication under a one-step delay, using the delayed information pattern during both training and evaluation. Each agent uses messages from the preceding timestep. We compare this model with the no-communication B3C backbone and immediate CODA on five settings. The communication variants select α separately from the same grid. Table 4 reports the resulting returns; implementation details are provided in Appendix H.

Delayed CODA preserves a substantial communication gain on the two HalfCheetah mixture settings and predator–prey, while its performance is close to the no-communication baseline on HC-k0-e and Sw-k0-e-m2. Compared with immediate communication, the delay has little effect on the HalfCheetah settings but removes most of the gain on Sw-k0-e-m2. Because the delayed model is trained and evaluated with delayed messages, this comparison assesses learning with delay rather than tolerance to an unexpected deployment-time delay.

Table 4: Performance of CODA with immediate and delayed communication, compared with no communication.

Setting	No comm. (B3C)	CODA (delay)	CODA
HC-k0-e	1396.4 ± 4.3	1388.9 ± 6.6	1392.2 ± 7.6
HC-k1-e-m1	2509.1 ± 63.4	3666.1 ± 27.5	3742.6 ± 35.8
HC-k1-e-m2	1398.8 ± 51.0	2790.9 ± 559.3	2855.0 ± 358.0
Sw-k0-e-m2	182.3 ± 29.4	180.0 ± 42.9	364.5 ± 36.2
PP0.5-e-m1	104.0 ± 6.8	128.3 ± 19.3	142.6 ± 23.4

Additional analyses examine the information carried by learned messages in Appendix E and alternative communication architectures in Appendix I.

5 Conclusion

We propose CODA, which learns deployment-time communication from offline multi-agent data collected without communication. Our analysis characterizes communication as a trade-off between information gain and the additional cost of learning a message-conditioned actor, and provides a condition under which communication tightens an overestimation bound. Empirically, CODA yields broad performance gains with B3C in partially observable tasks, while improvements with OMIGA are less consistent, showing that the benefit of communication depends on the underlying offline learning algorithm. Further analyses show that the gains are often accompanied by smaller policy–data action deviation and, in several settings, reduced empirical overestimation, while parameter-matched comparisons indicate that additional model capacity alone does not explain the gains. Finally, CODA remains effective under delayed communication in several tasks, showing that useful communication can still be learned when current messages are not immediately available.

Limitations and Future Work. Our theoretical analysis is developed under assumptions that capture how communication can reduce uncertainty about dataset-supported actions and affect offline overestimation. These assumptions do not cover all practical multi-agent settings, and the benefit of communication can vary across environments, datasets, and offline learning algorithms. Practical factors beyond the current scope include more complex dataset distributions, different levels of observability, and constrained or unreliable communication. Future work should extend the analysis to broader conditions and study how these factors shape the benefits of learned communication.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript writing and editing, refinement and checking of theoretical arguments, experimental-design feedback, result interpretation, and analysis/figure scripting. All AI-assisted outputs were reviewed and verified by the authors, who take responsibility for the final manuscript.

REFERENCES

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

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

Yali Du, Bo Liu, Vincent Moens, Ziqi Liu, Zhicheng Ren, Jun Wang, Xu Chen, and Haifeng Zhang. Learning correlated communication topology in multi-agent reinforcement learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 456–464, 2021.

J. Alberto Espinosa, F. Javier Lerch, and Robert E. Kraut. Explicit versus implicit coordination mechanisms and task dependencies: One size does not fit all. In Eduardo Salas and Stephen M. Fiore (eds.), *Team Cognition: Understanding the Factors that Drive Process and Performance*, pp. 107–129. American Psychological Association, 2004. doi: 10.1037/10690-006.

Yuchen Fang, Zhenggang Tang, Kan Ren, Weiqing Liu, Li Zhao, Jiang Bian, Dongsheng Li, Weinan Zhang, Yong Yu, and Tie-Yan Liu. Learning multi-agent intention-aware communication for optimal multi-order execution in finance. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4003–4012, 2023.

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

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. *Advances in neural information processing systems*, 30, 2017.

Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation. *Advances in neural information processing systems*, 31, 2018.

Jiechuan Jiang and Zongqing Lu. Offline decentralized multi-agent reinforcement learning. In *ECAI 2023*, 2023. doi: 10.3233/FAIA230390.

Seth Karten, Mycal Tucker, Siva Kailas, and Katia Sycara. Towards true lossless sparse communication in multi-agent systems. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7191–7197. IEEE, 2023.

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

Jeonghye Kim, Suyoung Lee, Woojun Kim, and Youngchul Sung. Decision convformer: Local filtering in metaformer is sufficient for decision making. In *The Twelfth International Conference on Learning Representations*, 2024a.

Jeonghye Kim, Suyoung Lee, Woojun Kim, and Youngchul Sung. Adaptive Q-aid for conditional supervised learning in offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024b.

Woojun Kim and Youngchul Sung. An adaptive entropy-regularization framework for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 16829–16852. PMLR, 2023.

Woojun Kim and Katia P. Sycara. B3C: A minimalist approach to offline multi-agent reinforcement learning. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems*, 2026. doi: 10.65109/UOYL1486.

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

Woojun Kim, Jongeui Park, and Youngchul Sung. Communication in multi-agent reinforcement learning: Intention sharing. In *International Conference on Learning Representations*, 2021.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Thomas W Malone and Kevin Crowston. What is coordination theory and how can it help design cooperative work systems? In *Proceedings of the 1990 ACM conference on Computer-supported cooperative work*, pp. 357–370, 1990.

Thomas W Malone and Kevin Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys (CSUR)*, 26(1):87–119, 1994.

Linghui Meng, Muning Wen, Chenyang Le, Xiyun Li, Dengpeng Xing, Weinan Zhang, Ying Wen, Haifeng Zhang, Jun Wang, Yaodong Yang, and Bo Xu. Offline pre-trained multi-agent decision transformer. *Machine Intelligence Research*, 20(2):233–248, 2023. doi: 10.1007/s11633-022-1383-7.

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

Frans A Oliehoek. Decentralized pomdps. In *Reinforcement learning: state-of-the-art*, pp. 471–503. Springer, 2012.

Ling Pan, Longbo Huang, Tengyu Ma, and Huazhe Xu. Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification. In *International conference on machine learning*, pp. 17221–17237. PMLR, 2022.

Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. FACMAC: Factored multi-agent centralised policy gradients. *Advances in Neural Information Processing Systems*, 34:12208–12221, 2021.

Jianzhun Shao, Yun Qu, Chen Chen, Hongchang Zhang, and Xiangyang Ji. Counterfactual conservative q learning for offline multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Junjie Sheng, Xiangfeng Wang, Bo Jin, Junchi Yan, Wenhao Li, Tsung-Hui Chang, Jun Wang, and Hongyuan Zha. Learning structured communication for multi-agent reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 36(2):50, 2022.

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

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

Wei-Cheng Tseng, Tsun-Hsuan Johnson Wang, Yen-Chen Lin, and Phillip Isola. Offline multi-agent reinforcement learning with knowledge distillation. *Advances in Neural Information Processing Systems*, 35:226–237, 2022.

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

Xiangsen Wang, Haoran Xu, Yinan Zheng, and Xianyuan Zhan. Offline multi-agent reinforcement learning with implicit global-to-local value regularization. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Haoran Xu, Li Jiang, Jianxiong Li, Zhuoran Yang, Zhaoran Wang, Victor Wai Kin Chan, and Xianyuan Zhan. Offline rl with no ood actions: In-sample learning via implicit value regularization. In *International Conference on Learning Representations*, 2023.

Yiqin Yang, Xiaoteng Ma, Chenghao Li, Zewu Zheng, Qiyuan Zhang, Gao Huang, Jun Yang, and Qianchuan Zhao. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:10299–10312, 2021.

Zhengbang Zhu, Minghuan Liu, Liyuan Mao, Bingyi Kang, Minkai Xu, Yong Yu, Stefano Ermon, and Weinan Zhang. MADiff: Offline multi-agent learning with diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

A Proofs and Detailed Theoretical Analysis

This appendix contains the full statements of the assumptions, the derivations, and the proofs of the results presented in Sec. 3.3.

We provide a simple analysis explaining when communication can tighten an overestimation bound in offline multi-agent RL. The key intuition is that, under partial observability, messages from other agents can act as proxy information about the latent context that determines which actions are well supported by the offline dataset. At the same time, conditioning the actor on messages enlarges the policy space, which may increase offline distribution shift.

Setup. We consider the decision problem from the perspective of agent i. Let O^i be its local observation, M^{-i} the messages received from the other agents, and U^i a latent context that determines which actions are well supported by the offline dataset (e.g., the global state, other agents' observations, or the trajectory-quality mode of the episode). The actor information is

$$Z_0^i = O^i, \qquad Z_c^i = (O^i, M^{-i}), \tag{7}$$

without and with communication, respectively. We assume that messages affect policy evaluation through the actor-induced action $A_\pi^i = \pi^i(Z^i)$, while the critic does not directly take messages as input. We analyze a fixed learned communication map and deterministic action maps $\pi^i(Z^i)$; the corresponding learned actor is denoted π_Z^i in the main text. All expectations in the comparison are under one common reference law ν on the contexts, not separate policy-dependent occupancy distributions. The other agents' policies and the remaining evaluation convention are held fixed in this one-agent comparison. The dependence of $\widehat{Q}$ and Q^π on the actor being evaluated is left implicit; these are not identified with a factorized critic's local Q_i head. Throughout this section, $\|\cdot\|$ denotes the Euclidean norm and, for a random vector X, we write $\mathrm{Var}(X \mid Z) = \mathrm{tr}(\mathrm{Cov}(X \mid Z))$ for its total conditional variance. Let

$$\bar{a}_D^i(u) = \mathbb{E}_D[A^i \mid U^i = u] \tag{8}$$

denote the dataset action center conditioned on the latent context. This quantity is used only as an analytical device; it does not require the behavior policy to observe U^i, and it serves as an oracle support-anchor analogous to the role played by behavior-policy actions in standard behavior-regularized offline RL (Kumar et al., 2020; Fujimoto & Gu, 2021). The anchor must be well-defined ν-almost surely. For example, take the reference law to be the dataset-context law, or require the U^i marginal under ν to be absolutely continuous with respect to the dataset U^i marginal. Assume $\mathbb{E}_D\|A^i\|^2 < \infty$, $\mathbb{E}_\nu\|\bar{a}_D^i(U^i)\|^2 < \infty$, and $\mathbb{E}_\nu\|\pi^i(Z^i)\|^2 < \infty$ for each actor compared. The subscript on $\mathbb{E}_D$ distinguishes the law defining the anchor from the common reference law used elsewhere.

Assumptions. We use two explicit modeling assumptions. First, positive critic error is bounded in terms of deviation from the dataset action center.

Assumption 1 (Support-controlled extrapolation error)**.** *For each of the two actors being compared, and its policy-induced action a^i, the following holds ν-almost surely:*

$$\left(\widehat{Q}(s,o,a^i) - Q^\pi(s,o,a^i)\right)_+ \le \kappa\,\|a^i - \bar{a}_D^i(U^i)\| + \epsilon_Q, \tag{9}$$

where $\kappa > 0$ and $\epsilon_Q \ge 0$ are finite constants shared by both actors, and $(x)_+ = \max\{x, 0\}$.

This assumption follows the offline-RL intuition that behavior-related constraints can limit extrapolation (Kumar et al., 2020; Fujimoto & Gu, 2021). The particular bound above, however, is an assumption of this analysis, not a consequence established by those works. A conditional action mean can lie between modes of the behavior distribution, so distance to $\bar{a}_D^i(U^i)$ is not, in general, a distance to the full dataset action support.

Second, the learned actor approximately minimizes support deviation given its information.

Assumption 2 (Approximate support-compatible actor)**.** *For $Z^i \in \{Z_0^i, Z_c^i\}$,*

$$\mathbb{E}\big[\|\pi^i(Z^i) - \bar{a}_D^i(U^i)\|^2\big] \le \min_f \mathbb{E}\big[\|f(Z^i) - \bar{a}_D^i(U^i)\|^2\big] + \epsilon_\pi(Z^i), \tag{10}$$

where the minimum is over all square-integrable measurable functions of Z^i, rather than a restricted neural-network class. The finite quantity $\epsilon_\pi(Z^i) \ge 0$ captures optimization error, approximation error, and additional deviation from the anchor predictor induced by the learned actor.

The term $\epsilon_\pi(Z_c^i)$ may be larger than $\epsilon_\pi(Z_0^i)$ because conditioning the actor on additional inputs (the messages M^{-i}) enlarges the policy space and may amplify offline learning error. The difference can also be zero or negative; enlarging a function class does not by itself force its best approximation error to increase. Behavior regularization, such as the BC loss in FACMAC+B3C+CODA or the behavior-related objective in OMIGA+CODA, motivates controlling the learned actor's deviation. We do not derive a rate relating this error to capacity, sample size, or regularization strength, nor prove that these algorithms keep $\epsilon_\pi(Z_c^i)$ close to $\epsilon_\pi(Z_0^i)$.

Information refinement. Define the residual support uncertainty under actor information Z^i as

$$\mathcal{U}(Z^i) = \mathbb{E}\big[\mathrm{Var}\big(\bar{a}_D^i(U^i) \mid Z^i\big)\big], \tag{11}$$

where $\mathrm{Var}(\cdot \mid Z^i)$ is the total conditional variance defined above.

Proposition 1 (Communication reduces residual support uncertainty)**.** *Since $Z_c^i = (O^i, M^{-i})$ refines $Z_0^i = O^i$,*

$$\mathcal{U}(Z_0^i) - \mathcal{U}(Z_c^i) = \mathbb{E}\big[\mathrm{Var}\big(\mathbb{E}[\bar{a}_D^i(U^i) \mid Z_c^i] \,\big|\, O^i\big)\big] \geq 0. \tag{12}$$

The gap vanishes when M^{-i} is conditionally independent of $\bar{a}_D^i(U^i)$ given O^i. More generally, it is zero exactly when $\mathbb{E}[\bar{a}_D^i(U^i) \mid Z_c^i] = \mathbb{E}[\bar{a}_D^i(U^i) \mid O^i]$ almost surely. Thus it captures action-center-relevant information revealed beyond the local observation; it need not be large.

Proof. The variance decomposition follows from the law of total variance applied to the nested information sigma-algebras $\sigma(O^i) \subseteq \sigma(O^i, M^{-i})$, componentwise for vector actions and then summed. Equivalently, the variance gap is

$$\mathbb{E}\Big[\big\|\mathbb{E}[\bar{a}_D^i(U^i) \mid Z_c^i] - \mathbb{E}[\bar{a}_D^i(U^i) \mid O^i]\big\|^2\Big].$$

The final statement follows because if $M^{-i} \perp \bar{a}_D^i(U^i) \mid O^i$, then $\mathbb{E}[\bar{a}_D^i(U^i) \mid O^i, M^{-i}] = \mathbb{E}[\bar{a}_D^i(U^i) \mid O^i]$, making the variance gap zero. □

Overestimation bound. We now relate residual support uncertainty to offline overestimation. Define

$$\mathrm{Over}(\pi_Z^i) = \mathbb{E}\left[\Big(\widehat{Q}(s, o, \pi^i(Z^i)) - Q^\pi(s, o, \pi^i(Z^i))\Big)_+\right]. \tag{13}$$

Theorem 1 (Information-induced support stabilization)**.** *Under Assumptions 1 and 2,*

$$\mathrm{Over}(\pi_Z^i) \leq \kappa\sqrt{\mathcal{U}(Z^i) + \epsilon_\pi(Z^i)} + \epsilon_Q. \tag{14}$$

Proof. By Assumption 1 and Jensen's inequality,

$$\mathrm{Over}(\pi_Z^i) \leq \kappa\sqrt{\mathbb{E}\big[\|\pi^i(Z^i) - \bar{a}_D^i(U^i)\|^2\big]} + \epsilon_Q. \tag{15}$$

By the standard fact that $f^*(z) = \mathbb{E}[\bar{a}_D^i(U^i) \mid Z^i = z]$ is the minimizer of the squared-error objective, we have the identity

$$\min_f \mathbb{E}\big[\|f(Z^i) - \bar{a}_D^i(U^i)\|^2\big] = \mathbb{E}\big[\mathrm{Var}\big(\bar{a}_D^i(U^i) \mid Z^i\big)\big] = \mathcal{U}(Z^i). \tag{16}$$

For completeness, this identity follows from the orthogonal decomposition

$$\begin{aligned}&\mathbb{E}\big[\|f(Z^i) - \bar{a}_D^i(U^i)\|^2\big] \\ &\quad = \mathcal{U}(Z^i) + \mathbb{E}\Big[\big\|f(Z^i) - \mathbb{E}[\bar{a}_D^i(U^i) \mid Z^i]\big\|^2\Big].\end{aligned} \tag{17}$$

The cross term is zero by conditioning on Z^i. Taking the conditional mean as f gives the minimum; taking $f = \pi^i$ shows that Assumption 2 is equivalently a bound on the learned actor's squared distance from this conditional mean. Combining the minimum-risk identity with Assumption 2 yields the stated bound. □

Corollary 1 (When communication tightens the bound)**.** *The communication-conditioned actor has a no larger overestimation bound than the non-communicating actor whenever*

$$\underbrace{\mathcal{U}(Z_0^i) - \mathcal{U}(Z_c^i)}_{\textit{information gain}} \;\geq\; \underbrace{\epsilon_\pi(Z_c^i) - \epsilon_\pi(Z_0^i)}_{\textit{actor-class cost}}. \tag{18}$$

Proof. Since $\sqrt{\cdot}$ is monotonically increasing and the two bounds in Theorem 1 share the same positive multiplier κ and additive constant ϵ_Q, the bound for Z_c^i is no larger than that for Z_0^i if and only if $\mathcal{U}(Z_c^i) + \epsilon_\pi(Z_c^i) \leq \mathcal{U}(Z_0^i) + \epsilon_\pi(Z_0^i)$, which rearranges to the stated condition. The information-gain term is non-negative by Proposition 1. This orders the displayed upper bounds, not necessarily the actual overestimation errors or the returns of the two policies. □

Implications. The analysis identifies a possible benefit of communication when messages reduce uncertainty about dataset-supported action centers, especially when the local observation O^i alone leaves substantial residual uncertainty $\mathcal{U}(Z_0^i)$. Partial observability by itself does not guarantee a large information gain. Other agents' messages must resolve uncertainty relevant to $\bar{a}_D^i(U^i)$. If this center is already determined by O^i, the gain is zero. For example, full state observability implies zero residual uncertainty when U^i is just that observed state, but not necessarily when it also contains an unobserved behavior mode.

Behavior regularization may help control the actor-class cost. Under the stated assumptions, information gain exceeding that cost gives a tighter overestimation bound. This motivates separate empirical diagnostics of action-relevant predictability, policy–data action deviation, and positive critic–return gaps. Logged actions are not the oracle conditional center, and joint rollout diagnostics are not the per-agent reference-law quantity analyzed here. These measurements therefore test related predictions rather than establish that every instance of communication reduces actual overestimation or improves return.

Scope for stochastic actors. The proof above concerns deterministic action maps. For an action sampled conditionally on Z^i with randomness independent of the latent context given Z^i, the same decomposition has an additional conditional action-variance term. That term must be included in the actor error when applying an analogous bound to stochastic actions. The deterministic argument alone does not identify our sampled-action training critic with the return of a deterministic rollout policy.

Connection to the predictive probe (additional analysis). The empirical probes predict A^i, not $X = \bar{a}_D^i(U^i)$. In this paragraph Z denotes Z_0^i or Z_c^i. For a fixed representation and dataset law, a separate least-squares identity is

$$L(g_Z) = \mathbb{E}_D \left\| A^i - g_Z(Z) \right\|^2 = \mathbb{E}_D[\mathrm{Var}(A^i \mid Z)] + e_Z, \qquad e_Z = \mathbb{E}_D \left\| g_Z(Z) - \mathbb{E}_D[A^i \mid Z] \right\|^2.$$

Consequently, at the population level, with all probe variables square-integrable,

$$L(g_0) - L(g_c) = \underbrace{\mathbb{E}_D[\mathrm{Var}(A^i \mid O^i)] - \mathbb{E}_D[\mathrm{Var}(A^i \mid O^i, M^{-i})]}_{\text{nonnegative Bayes action-prediction gain}} + e_0 - e_c.$$

A fitted finite-sample probe may have a negative empirical gain even though the Bayes term is nonnegative. Its error terms are not the actor's ϵ_π, and the Bayes term for A^i is not automatically the theorem's anchor uncertainty. This is why we use the probe as a related diagnostic rather than a direct estimator of Eq. (3).

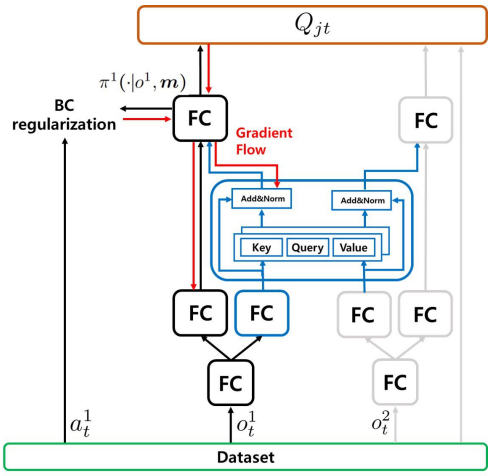


Figure 4: Policy and communication-policy structure in FACMAC+B3C+CODA for two agents. The actor and communication modules are trained through gradients from the RL objective and BC regularization. The diagram is schematic; parameter sharing is described in the text.

B Communication Policy and Implemented Objectives

In the integration of communication with both OMIGA and FACMAC+B3C, each agent has an action policy and a communication policy. The communication policy is conditioned on its own observation, and the action policy depends on that observation and the exchanged messages. This connection between communication policies and other agents' action policies enables gradient flow, allowing them to be trained jointly. Building on previous research in multi-agent communication (Jiang & Lu, 2018; Niu et al., 2021), we use an attentional communication network based on multi-head attention (Vaswani et al., 2017).

B.1 Message generation, attention, and actions

Each agent encodes its local observation and generates an outgoing message before information is aggregated across agents. The no-delay implementation shares actor parameters and appends a one-hot agent identity e_i to its observation. The local encoding and outgoing message are

$$h_i = \mathrm{ReLU}(W_1[o^i, e_i] + b_1), \qquad m^i = W_m h_i + b_m,$$

where h_i has dimension 256 and m^i has dimension 8. The messages are fed into a self-attention module whose query, key, and value components are learned projections. For each head,

$$\mathrm{Attention}(Q, K, V) = \mathrm{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V, \tag{19}$$
$$Q = \{W_q m^i\}_{i=1}^N, \quad K = \{W_k m^i\}_{i=1}^N, \quad V = \{W_v m^i\}_{i=1}^N.$$

The four-head implementation uses $d_k = d_v = 8$ per head. The output projection is followed by a residual connection and layer normalization. The resulting agent-specific representation is concatenated with h_i and passed through a second 256-unit ReLU layer before the mean and log-standard-deviation heads. Actions follow a tanh-squashed Gaussian, with deterministic evaluation using its squashed mean. Self-attention includes the agent's own message; the information probes explicitly exclude that message.

Each outgoing message depends only on its sender's observation and identity, whereas the attention output depends on all agents' messages. Communication parameters receive gradients through receiver actors without supervised target messages.

B.2 B3C REDUCTION, NORMALIZATION, AND CLIPPING

For a sampled minibatch $\mathcal{B}$ and reparameterized actor actions $\tilde{\boldsymbol{a}}_b$, the no-delay implementation uses

$$\bar{Q}_{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} \widehat{Q}_{jt}(s_b, \boldsymbol{o}_b, \tilde{\boldsymbol{a}}_b), \qquad c_\alpha = \frac{\alpha}{\mathrm{sg}(|\bar{Q}_{\mathcal{B}}| + 10^{-8})}.$$

Thus the actor minimizes $-c_\alpha \bar{Q}_{\mathcal{B}} + \beta \ell_{\mathrm{BC}}$, where ℓ_{BC} averages squared errors over batch, agent, and action coordinates. The normalization is the absolute *minibatch mean*, not the mean of absolute Q values, and the normalizer is detached. We specify this reduction because the two normalizations are not interchangeable.

For a transition mask d_b^{cont}, the critic target is

$$y_b = r_b + \gamma d_b^{\mathrm{cont}} \min\{\widehat{Q}_{\mathrm{target}}(s'_b, \boldsymbol{o}'_b, \tilde{\boldsymbol{a}}'_{\pi,b}),\ MR_{\max}\},$$

where $R_{\max}$ is the maximum dataset episode return. In the no-delay code, both the next policy action and the actor-loss action are sampled; the target critic is updated by a Polyak average. Clipping applies to the bootstrapped Q term, before adding the immediate reward. The PO experiments use $M = 1$ and $\beta = 1$. The fully observable settings are listed separately in Sec. C.3.

B.3 OMIGA EXTRACTION

OMIGA (Wang et al., 2023) linearly decomposes its joint Q- and V-functions as $Q_{jt}(\boldsymbol{o}, \boldsymbol{a}) = \sum_i w_i(\boldsymbol{o}) Q_i(o^i, a^i) + b(\boldsymbol{o})$ and $V_{jt}(\boldsymbol{o}) = \sum_i w_i(\boldsymbol{o}) V_i(o^i) + b(\boldsymbol{o})$, using shared weight and bias functions. Before adding communication, the actor objective is

$$\max_{\pi^i} \mathbb{E}_{\mathcal{D}} \left[\exp\left(\frac{w_i(\boldsymbol{o})\big(Q_i(o^i, a^i) - V_i(o^i)\big)}{\alpha} \right) \log \pi^i(a^i \mid o^i) \right]. \tag{20}$$

CODA conditions this likelihood on the exchanged messages and learns the communication module through the same objective. The implemented actor weight is

$$\omega_i = \mathrm{sg}\left[\exp\left(\mathrm{clip}\left(\frac{w_i(\boldsymbol{o})(Q_{i,\mathrm{target}}(o^i, a^i) - V_i(o^i))}{\alpha}, -10, 10 \right) \right) \right].$$

The actor minimizes $-\operatorname{mean}_i \mathbb{E}_{\mathcal{B}}[\omega_i \log \pi^i(a^i \mid o^i, m)]$. Removing communication removes m from the input. The no-delay Q/V updates do not backpropagate through the actor. The native Q is learned with a bootstrapped V target, while the actor is extracted with weighted likelihood. Its native value is therefore not automatically an unregularized evaluator of the final extracted policy. This distinction is used when interpreting the OMIGA diagnostics.

C Experimental Protocols and Hyperparameters

C.1 Reporting blocks

Table 5: Experimental blocks and run counts.

Block	Matrix / reporting rule	Runs
PO B3C grid	21 settings × 2 methods × 5 α × 5 seeds	1050
PO OMIGA grid	21 settings × 2 methods × 2 α × 5 seeds	420
Attribution	8 settings × 4 methods × 10 seeds; frozen α	320
Selected-α diagnostics	4 settings × 2 methods × 5 seeds	40
High-α pilot	3 settings × 2 methods × 3 seeds	18
OMIGA diagnostic	HC-k1-e-m1 × 2 methods × 5 seeds	10
Probe representations	4 settings × 3 CODA seeds	12
Delayed CODA	5 settings × 5 α × 5 seeds	125

The selected-α diagnostic block compares B3C and B3C+CODA at a common α within each setting: 1 for HC-k1-e-m1 and HC-k0-e, and 0.5 for HC-k1-e-m2 and Sw-k0-m1-m2. The probe analysis uses three independently trained CODA models per setting.

C.2 Training and evaluation

Table 6 gives the selected α for each method in the PO grid. The PO B3C/CODA grid uses batch size 32, no delay, message dimension 8, $\beta = 1$, and $M = 1$. MuJoCo/Swimmer training budgets are 1.5 million updates, with 30 scheduled evaluations at $0, 50000, \ldots, 1450000$; the last six form the final-20% summary. Predator–Prey uses 200000 updates with a last scheduled evaluation at 199000; its final 20% rule is applied to its own evaluation schedule. For each grid cell, uncertainty is computed over five per-run summaries, not over individual checkpoints. Alpha selection maximizes the five-seed late mean, with ties favoring the smaller α. This is selection on environment evaluation returns and is not a claim of validation-free tuning.

Table 6: Selected α in the PO grid. Each base algorithm and its CODA extension are tuned separately.

	OMIGA		B3C	
Setting	Base	+CODA	Base	+CODA
HC-k0-e	10	1	0.5	1
HC-k0-m1	1	1	4	4
HC-k0-m2	10	1	8	8
HC-k0-e-m1	1	1	4	2
HC-k0-e-m2	1	1	4	8
HC-k0-m1-m2	1	1	8	8
HC-k1-e	10	10	2	2
HC-k1-m1	1	1	1	1
HC-k1-m2	1	10	8	8
HC-k1-e-m1	1	1	1	1
HC-k1-e-m2	1	1	0.5	0.5
HC-k1-m1-m2	1	1	2	1
Sw-k0-e	10	1	0.5	0.5
Sw-k0-m1	10	1	0.5	0.5
Sw-k0-m2	10	1	1	0.5
Sw-k0-e-m1	10	1	1	2
Sw-k0-e-m2	1	1	2	0.5
Sw-k0-m1-m2	1	10	0.5	0.5
PP0.5-e	10	10	0.5	0.5
PP0.5-m1	1	1	0.5	2
PP0.5-e-m1	10	1	1	0.5

OMIGA uses batch size 128, learning rate 5×10^{-4}, discount 0.99, target update 0.005, 256-unit policy/Q/V hidden layers, 64-unit mixer hidden size, gradient-norm limit 1, and no target clipping. Its grid is $\{1, 10\}$. The two integrations share their corresponding backbone's training protocol, but the B3C and OMIGA grids have different computational budgets.

The mechanism block fixes the preselected CODA α for both methods, uses 32 deterministic evaluation episodes per checkpoint, and reports the predefined six-checkpoint interval $[1.20\text{M}, 1.45\text{M}]$ as well as the final checkpoint. It uses discount 0.99, a fixed 10000-transition offline subset for action-deviation measurements, and the same B3C/CODA non-α settings as the PO performance grid. The fully observable experiments use the hyperparameters listed in Sec. C.3.

C.3 Hyperparameters for the fully observable experiments

Table 7 lists the CODA hyperparameters used for the fully observable experiments. The PO hyperparameters are specified separately in Table 6. For the fully observable experiments, $M = 0.1$ is used

Table 7: CODA hyperparameters for the fully observable experiments.

Environment	Dataset	α	M
Hopper	expert	16	1
	medium	8	1
	medium-replay	0.25	0.1
	medium-expert	1	1
Ant	expert	1	1
	medium	1	1
	medium-replay	1	1
	medium-expert	0.5	1
HalfCheetah	expert	16	1
	medium	16	1
	medium-replay	16	1
	medium-expert	16	1

Table 8: Actor parameter counts in the ten-seed attribution block. PM is the exact parameter-matched no-communication baseline.

Setting	B3C	PM	RawObs	B3C+CODA
HC-k1-e-m1	69,122	74,266	74,242	74,266
HC-k1-e-m2	69,122	74,266	74,242	74,266
HC-k1-m2	69,122	74,266	74,242	74,266
HC-k0-m1-m2	68,610	73,754	71,170	73,754
HC-k0-e	68,610	73,754	71,170	73,754
Sw-k0-e-m2	69,380	74,524	73,476	74,524
Sw-k0-m1-m2	69,380	74,524	73,476	74,524
PP0.5-e-m1	71,940	77,084	80,132	77,084

for Hopper medium-replay and $M = 1$ for the other settings in Table 7. The PO experiments use $M = 1$.

C.4 Capacity and observation-sharing comparisons

The parameter-matched model (PM) uses the same number of actor parameters as CODA and retains its attention-related capacity while applying attention separately to each agent's own message. RawObs concatenates the current raw local observations of all agents, in fixed order, with a receiver identity. The same frozen α, dataset, and training/evaluation schedule are used within each attribution setting. The final scheduled checkpoint, rather than a best checkpoint or the PO-grid late mean, is the attribution outcome. The fixed values are $\alpha = 1$ for HC-k1-e-m1, HC-k0-m1-m2, HC-k0-e, and Sw-k0-e-m2; $\alpha = 4$ for HC-k1-e-m2 and HC-k1-m2; $\alpha = 0.5$ for Sw-k0-m1-m2; and $\alpha = 8$ for PP0.5-e-m1. These common values apply to all four methods in Table 3 and differ from method-specific selection in Table 6.

D STATISTICAL REPORTING

The unit of replication is the training seed. For a late metric, we first average the specified checkpoints within a seed and then compute cross-seed means and sample SDs. An identical integer seed labels a pairing; it does not mean all random draws are identical between architectures. We report paired differences rather than count checkpoints as trials.

For the eight-setting attribution block, we use 50000 paired-bootstrap resamples and an exact two-sided sign-flip test on ten seed differences. The table below also reports Holm adjustment across these eight tests. Six individual bootstrap intervals exclude zero, whereas three adjusted tests are below 0.05; these are distinct statements. The interval and p-value procedures should not be conflated. Here exact refers to enumerating all sign flips; validity as a test also requires sign-exchangeable paired differences under the null.

Table 9: B3C+CODA minus B3C in the ten-seed attribution block. Bootstrap intervals and exact p-values are computed from the paired seed-level results; Holm adjustment is applied across eight tests.

Setting	Mean difference	Bootstrap 95% CI	Exact p	Holm p
HC-k1-e-m1	+1281.04	$[1227.29, 1336.69]$	0.0020	0.0156
HC-k1-e-m2	+1278.96	$[414.27, 2007.91]$	0.0195	0.0781
HC-k1-m2	+52.08	$[-77.43, 192.88]$	0.4941	0.9883
HC-k0-m1-m2	+93.29	$[61.91, 121.33]$	0.0039	0.0234
HC-k0-e	-2.37	$[-12.56, 7.49]$	0.6523	0.9883
Sw-k0-e-m2	+111.43	$[42.46, 181.58]$	0.0215	0.0781
Sw-k0-m1-m2	+96.94	$[87.52, 105.53]$	0.0020	0.0156
PP0.5-e-m1	+45.98	$[21.11, 69.86]$	0.0117	0.0586

With five seeds, the smallest nonzero two-sided sign-flip p-value is $2/2^5 = 0.0625$; with three it is $2/2^3 = 0.25$. These small blocks are used for effect sizes, individual-seed patterns, and descriptive diagnostics, not significance claims. The PO-grid selection and reporting use the same five seeds. Comparisons aggregated across settings are descriptive because tasks, return scales, and overlapping mixture components do not form independent, identically distributed observations. We do not average raw return across tasks.

E Post-Training Analysis of Message Information

After CODA training is complete, we freeze its final encoders and fit separate action predictors. For each receiver i, the prediction target is its own logged action A^i, using its local observation O^i with or without the other agents' messages. These predictors are used only for analysis; their loss never updates CODA and is not part of policy training or deployment.

Local uses local observations and zero-filled message slots. *Correct* adds other agents' messages computed from the same sample before attention aggregation. *Shuffled* uses messages from another episode within the same data component, providing a reference for sample correspondence. The test episodes are held out from probe fitting, but not from encoder training.

Fig. 5 and Table 10 report the results. Correct reduces MSE by 52.0% and 56.3% relative to Local in HC-k1-e-m1 and HC-k1-e-m2, respectively, and also outperforms Shuffled. Neither $k = 0$ setting shows such an improvement. For each variant v, the probe loss is $L_v = \mathbb{E}_{\text{probe test}}\|A^i - g_v(Z_v^i)\|^2$, averaged over agents. This is an auxiliary prediction error, not a return metric, the actor's ϵ_π, or a direct measurement of $\mathcal{U}(Z_0) - \mathcal{U}(Z_c)$.

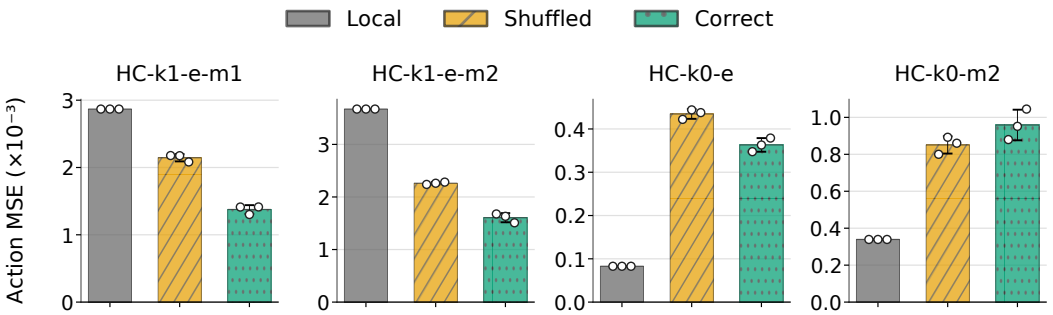


Figure 5: Post-training prediction of each receiver's own logged action from its observation and other agents' messages. Separate probes are fitted after CODA is frozen, using Local, within-component Shuffled, or sample-matched Correct inputs. Bars show mean ± SD across three CODA models; lower MSE is better. This is not a policy-training loss or return measure.

Table 10: Action-prediction MSE ($\times 10^{-3}$) is reported as mean ± SD across three frozen RL representations after averaging six agents. Local results coincide across RL seeds because its input, initialization, split, and training order are fixed. The split is probe-held-out only.

Setting	α	Local	Shuffled	Correct
HC-k1-e-m1	1	2.8681 ± 0.0000	2.1451 ± 0.0544	1.3760 ± 0.0645
HC-k1-e-m2	0.5	3.6721 ± 0.0000	2.2609 ± 0.0219	1.6057 ± 0.0870
HC-k0-e	1	0.0828 ± 0.0000	0.4347 ± 0.0113	0.3631 ± 0.0157
HC-k0-m2	8	0.3391 ± 0.0000	0.8509 ± 0.0468	0.9587 ± 0.0828

E.1 Frozen Representation and Data Access

The CODA α values are 1 for HC-k1-e-m1, 0.5 for HC-k1-e-m2, 1 for HC-k0-e, and 8 for HC-k0-m2. For seeds 0–2, the final scheduled actor/communication checkpoint at step 1450000 is frozen and verified. Each dataset contains 1000 trajectories of 1000 action transitions. Mixtures use 500 trajectories per source component. The local observation dimensions are 4 for the two $k = 1$ settings and 2 for the two $k = 0$ settings; each of six agents has one action coordinate.

For each receiver i, the communication feature is the ordered concatenation of the other five agents' eight-dimensional outgoing messages, before attention aggregation. The receiver's own message is excluded, giving a 40-dimensional input. The probe tests how well logged actions can be predicted from these messages, rather than what information remains after attention aggregation or what the final actor actually uses.

Important access distinction. The RL actors and encoders are trained on the full 1000-episode offline dataset. The subsequent 800/200 split prevents a trajectory from appearing in both *probe* training and probe testing, but those 200 episodes were not held out from RL representation learning. Accordingly, the results do not establish encoder generalization to previously unseen trajectories. The same caveat applies even though the encoder is frozen when the supervised probes are trained.

E.2 Split, shuffle, and equal-capacity fitting

A fixed episode-level split preserves component proportions by assigning 400/100 train/test trajectories per component in mixtures, or 800/200 for a single-source dataset. All variants and all RL seeds share that split. Within each split and component, a fixed episode derangement chooses a different donor episode; messages are computed at the same timestep and preserve sender identity. Train and test donors remain within their respective splits. Because component membership is preserved, the shuffle removes current-episode correspondence but not all dataset-mode or time-index information. It is not an unconditional independence baseline.

Each variant uses independent per-agent two-hidden-layer ReLU MLPs of width 256, implemented in a batched form. Input dimensions are 44 for $k = 1$ and 42 for $k = 0$ for all variants. Local fills the 40 standardized message slots with zeros. Nominal parameter counts therefore match, with 77569 per agent and variant for $k = 1$, and 77057 for $k = 0$. Extra active inputs can still change optimization difficulty, so equal counts do not eliminate all finite-probe effects.

The frozen protocol uses Adam at 10^{-3} for 20 epochs, batch size 4096 per agent, no weight decay, and no early stopping. Initialization and minibatch order are shared across variants using one fixed probe seed. Observation and message standardization statistics are computed from probe-training episodes only; action targets are not standardized. The evaluation covers all 200000 test transitions per agent. A finite training budget was prespecified; convergence to a Bayes predictor is not assumed.

E.3 Aggregation and interpretation

For each representation, MSE is first computed per agent, then macro-averaged over six agents. Means and sample SDs are then taken over three RL representations. The primary relative proxy is a ratio of setting means,

$$\widehat{\Delta}_{\mathrm{rel}} = \frac{\overline{L}_0 - \overline{L}_c}{\overline{L}_0}.$$

The local probe is identical across RL seeds, explaining its zero cross-RL-seed SD; this is not zero uncertainty about the population risk. Variation from dataset splitting and probe initialization is not assessed. Agent-seed outcomes are not 18 independent replications.

Correct beats both Local and Shuffled for each of the 18 agent–representation pairs in each $k = 1$ mixture. In each $k = 0$ setting, Correct beats Local in none of these 18 pairs. These patterns describe the prescribed finite decoder. In particular, worse Correct MSE does not prove redundant messages, zero mutual information, or zero oracle uncertainty reduction. The very small local MSE on HC-k0-e also makes relative negative percentages large; raw MSE must accompany any relative proxy. The design confounds observation setting with mixture composition, so it does not identify either as the cause of the contrast.

Table 11: Seed-level uncertainty for the seven B3C diagnostic conditions is reported as mean ± sample SD of per-seed late-window averages. Action L2-squared is displayed in units of 10^{-2}; the other columns use their raw units.

Setting	α	n	Method	Return	D_{act} ($\times 10^{-2}$)	$\mathbb{E}[\delta_+]$
HC-k1-e-m1	1	5	B3C	2530.4 ± 46.9	0.350 ± 0.032	83.49 ± 29.50
			B3C+CODA	3749.2 ± 46.4	0.174 ± 0.009	23.31 ± 5.28
HC-k1-e-m2	0.5	5	B3C	1412.5 ± 41.7	0.312 ± 0.020	83.73 ± 8.94
			B3C+CODA	2757.7 ± 387.6	0.123 ± 0.008	82.98 ± 27.42
Sw-k0-m1-m2	0.5	5	B3C	189.1 ± 10.7	0.595 ± 0.010	3.50 ± 0.54
			B3C+CODA	278.4 ± 6.2	0.170 ± 0.004	3.64 ± 0.13
HC-k0-e	1	5	B3C	1391.3 ± 8.4	0.024 ± 0.011	12.26 ± 1.40
			B3C+CODA	1392.3 ± 9.3	0.024 ± 0.007	13.23 ± 2.40
HC-k0-m1-m2	8	3	B3C	1115.7 ± 88.7	1.583 ± 0.159	24.97 ± 11.86
			B3C+CODA	1343.4 ± 6.8	1.268 ± 0.023	25.16 ± 1.44
HC-k0-m2	8	3	B3C	1172.4 ± 59.0	1.034 ± 0.060	20.89 ± 1.53
			B3C+CODA	1331.2 ± 63.5	1.405 ± 0.097	30.47 ± 14.69
HC-k1-m2	8	3	B3C	1867.1 ± 506.8	1.117 ± 0.979	74.29 ± 72.96
			B3C+CODA	1960.2 ± 604.2	1.032 ± 0.850	28.89 ± 7.65

F Action and Critic Diagnostics

F.1 Definitions and comparability

For a fixed offline subset $\mathcal{S}$ of 10000 transitions, we compute

$$D_{\text{act}} = \frac{1}{|\mathcal{S}|N} \sum_{b \in \mathcal{S}} \sum_{i=1}^{N} \left\| a^i_{\pi,b} - a^i_b \right\|^2, \qquad D_{\text{L2}} = \frac{1}{|\mathcal{S}|N} \sum_{b \in \mathcal{S}} \sum_{i} \left\| a^i_{\pi,b} - a^i_b \right\|.$$

The policy action is deterministic. These statistics compare to the particular logged action, not to a conditional action center, nearest supported mode, or support probability. Neither is an estimate of ϵ_π in the theorem.

For the rollout diagnostics, G_t is discounted return-to-go on the same evaluation episode whose state, observation, and policy action are input to the native joint critic. Signed, positive, and absolute gaps satisfy

$$\mathbb{E}[\delta] = \mathbb{E}[\delta_+] - \mathbb{E}[(-\delta)_+], \qquad \mathbb{E}[|\delta|] = \mathbb{E}[\delta_+] + \mathbb{E}[(-\delta)_+].$$

A lower signed gap can result from increased negative error rather than reduced positive error. For example, at the HC-k1-e-m2 selected operating point, the positive gap changes little ($83.73 \to 82.98$), while the mean negative part increases approximately from 1.36 to 34.45. A signed-gap decline alone is therefore insufficient evidence for smaller positive discrepancies.

These diagnostics have three further limitations. First, evaluation visits differ across policies, so a gap difference is not a controlled critic-only comparison. Second, the sampled actor used in B3C training and bootstrapping differs from the deterministic rollout actor; their equivalence is not assumed. OMIGA additionally learns a regularized value used for policy extraction, not necessarily the final actor's unregularized value. Third, the finite rollout ends with zero residual return, whereas training uses the stored continuation masks. The physical time-limit wrapper and the dataset-generation terminal convention must be distinguished. We report the finite-horizon evaluation protocol and do not certify these gaps as unbiased estimates of infinite-horizon Q^π error.

Even with a perfectly policy-matched unbiased Monte-Carlo target, taking a positive part before averaging introduces sensitivity to rollout randomness. Consequently, $\mathbb{E}[(\widehat{Q} - G)_+]$ is not generally identical to $\mathbb{E}[(\widehat{Q} - Q^\pi)_+]$. The latter is the theoretical quantity.

F.2 All seven B3C settings

Table 2 contains all seven primary setting means. The signed and absolute gaps are included below. The five-seed and three-seed blocks are reported separately rather than pooled. At each condition,

the six checkpoints in $[1.20\mathrm{M}, 1.45\mathrm{M}]$ are averaged within seed first. Final-checkpoint and raw checkpoint values remain available in the accompanying figure-input CSVs.

Table 12: Additional late critic–return gaps, B3C → B3C+CODA. Signed values are net biases, not an error magnitude with a universal lower-is-better direction.

Setting	α	n	Signed $\mathbb{E}[\delta]$	Absolute $\mathbb{E}[\lvert\delta\rvert]$
HC-k1-e-m1	1	5	$69.60 \to -37.00$	$97.37 \to 83.62$
HC-k1-e-m2	0.5	5	$82.37 \to 48.53$	$85.09 \to 117.43$
Sw-k0-m1-m2	0.5	5	$2.84 \to 3.64$	$4.15 \to 3.65$
HC-k0-e	1	5	$-1.34 \to 2.39$	$25.85 \to 24.07$
HC-k0-m1-m2	8	3	$20.99 \to 23.78$	$28.95 \to 26.54$
HC-k0-m2	8	3	$17.37 \to 28.79$	$24.40 \to 32.15$
HC-k1-m2	8	3	$69.17 \to 18.16$	$79.42 \to 39.61$

The high-α pilot was deliberately limited to three paired seeds. HC-k0-m1-m2 gains return and lowers action deviation in 3/3 pairs, but lowers positive gap in only 1/3. HC-k0-m2 raises return, action deviation, and positive gap in all three pairs. HC-k1-m2 is aligned in setting means, but the individual seeds do not support a uniform chain. One has lower critic/action gaps with lower return, one has higher return with larger gaps, and one aligns in all three. Averaging these patterns is not a mediation analysis. Fig. 10 shows the corresponding individual-seed trajectories.

Exploratory means over the complete 30-checkpoint trajectory also do not establish a universal positive-gap reduction. The selected-α means are approximately $98.67 \to 31.54$ for HC-k1-e-m1, $211.22 \to 174.36$ for HC-k1-e-m2, and $4.21 \to 4.18$ for Sw-k0-m1-m2. These full-training summaries were considered after the predefined late analysis and are reported as exploratory, not substituted as the primary endpoint.

F.3 OMIGA DIAGNOSTIC

At fixed $\alpha = 1$ in HC-k1-e-m1, the five-seed diagnostic gives mean late return $3466.1 \to 2553.4$ and action-deviation reduction of approximately 83.7%. Its native absolute critic–rollout gap decreases from about 114.3 to 64.7, while its positive gap increases from about 13.5 to 26.2. Thus improved logged-action fitting is not sufficient for better return, and the particular failure explanation based on increased measured action deviation is not supported. We leave the detailed interaction between communication, value regularization, and policy extraction unresolved; no OMIGA information probe or causal intervention result is asserted.

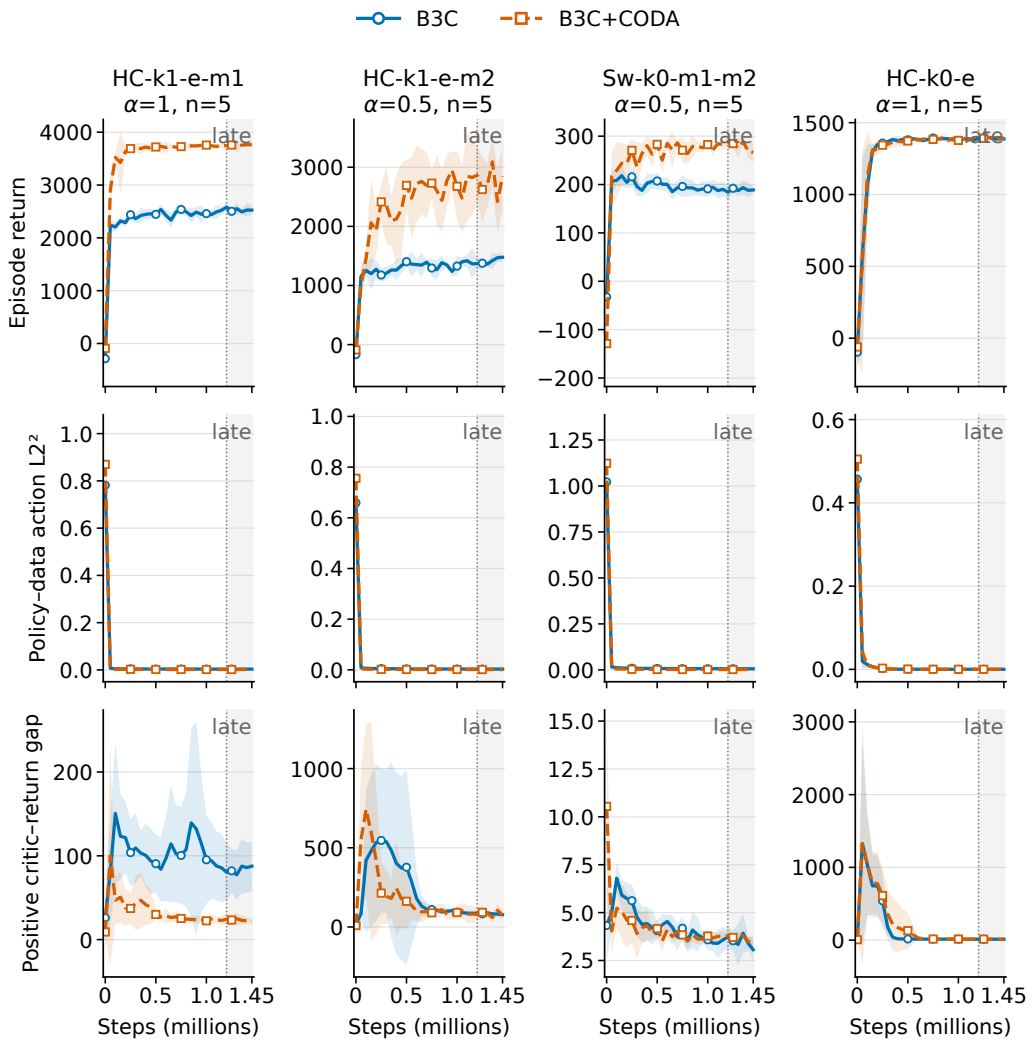


Figure 6: Complete selected-α training diagnostics. Each curve is a five-seed mean with SD; the same α is used for both methods in each setting.

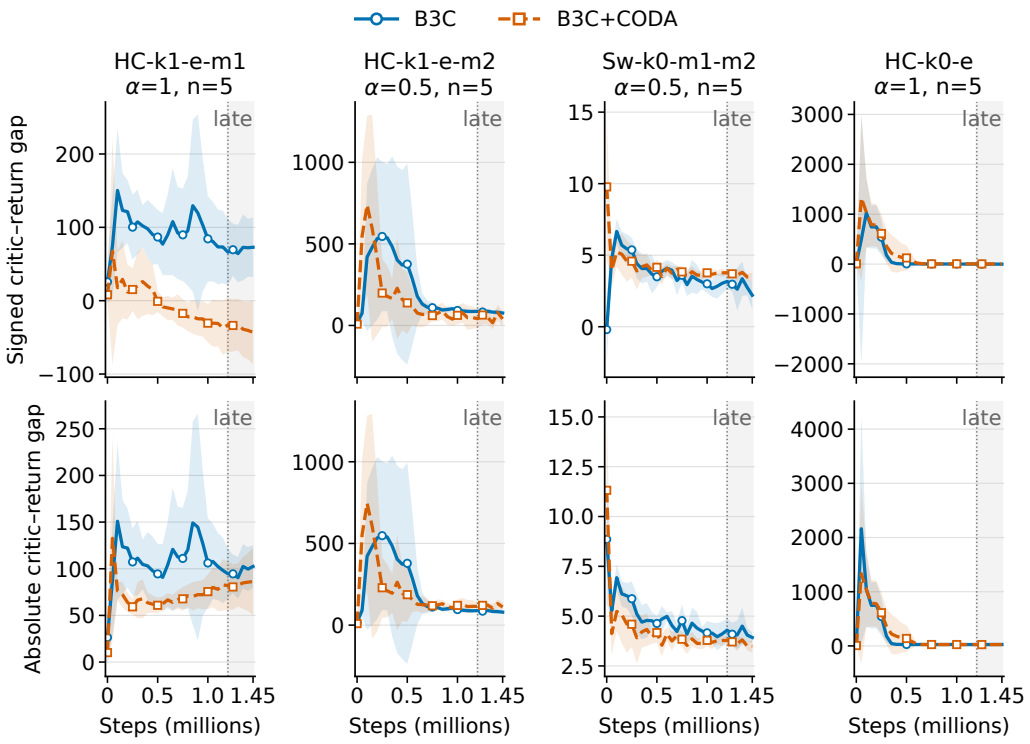


Figure 7: Signed and absolute gaps for the selected-α diagnostics. Curves show means with SD across five seeds; signed gaps have no universal lower-is-better interpretation.

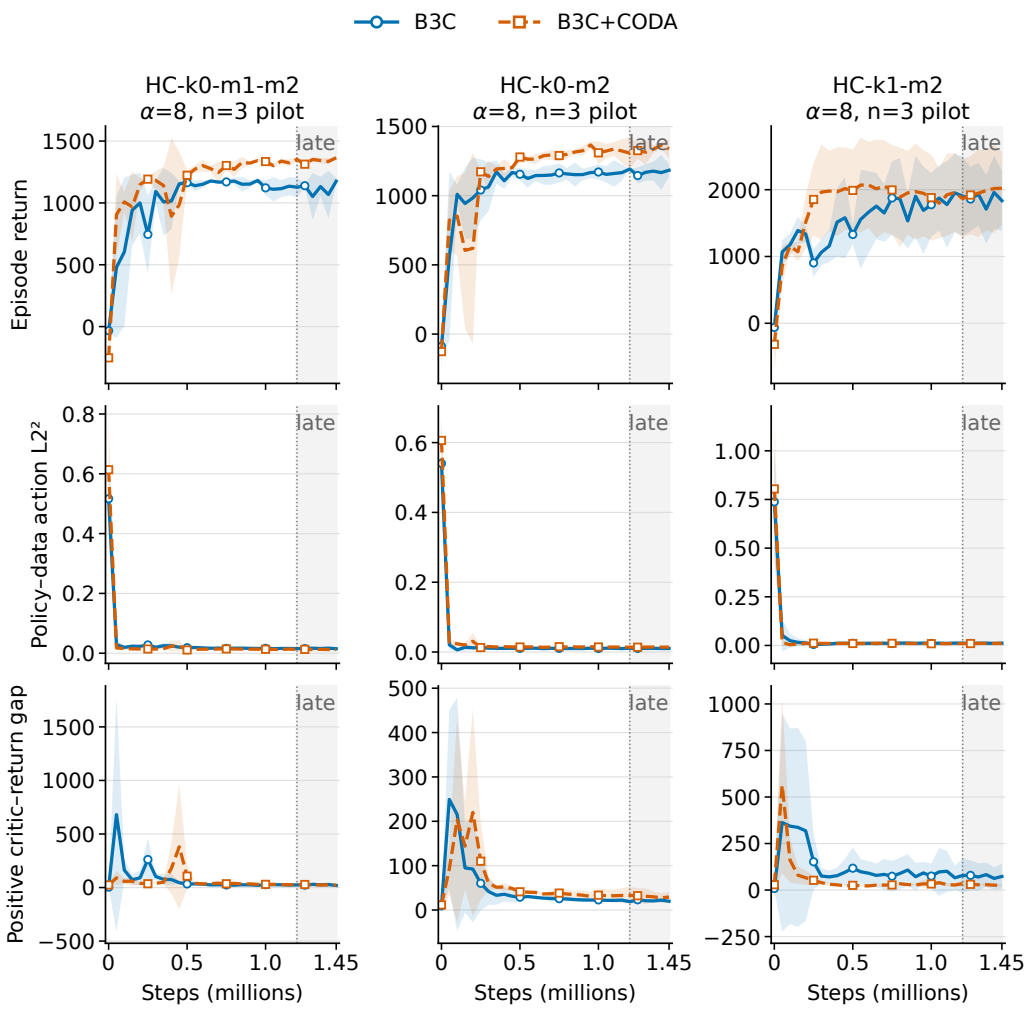


Figure 8: Complete high-α pilot diagnostics. Each curve is a three-seed mean with SD. These results are descriptive; the hyperparameters are fixed independently of these gaps.

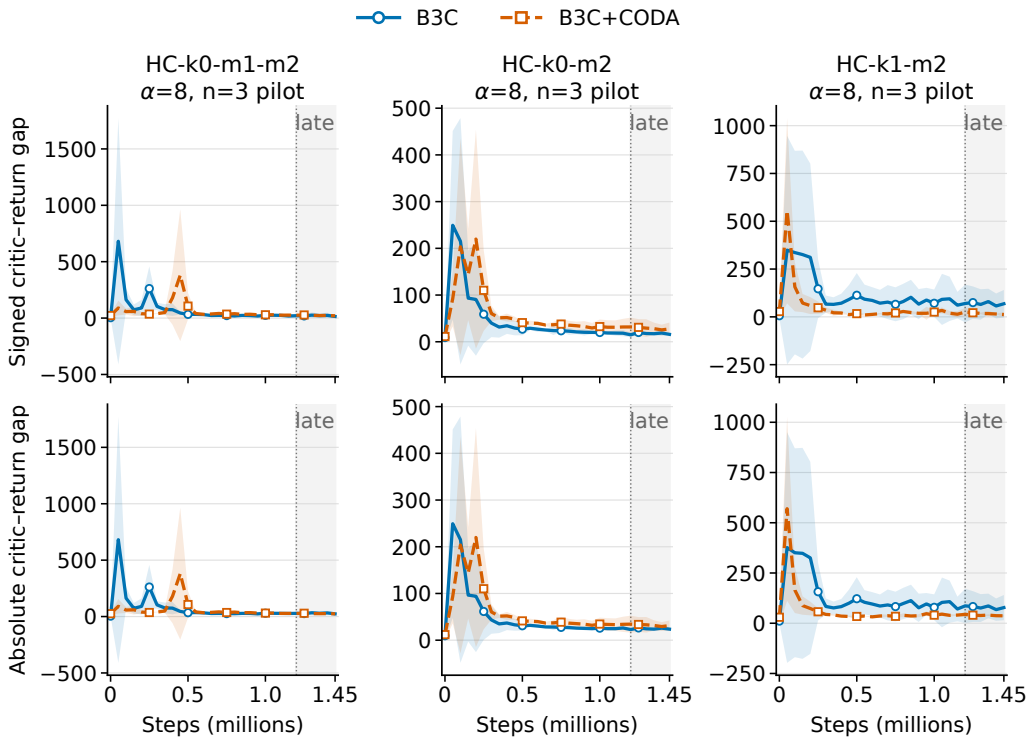


Figure 9: Signed and absolute gaps for the high-α pilot. Curves show means with SD across three seeds.

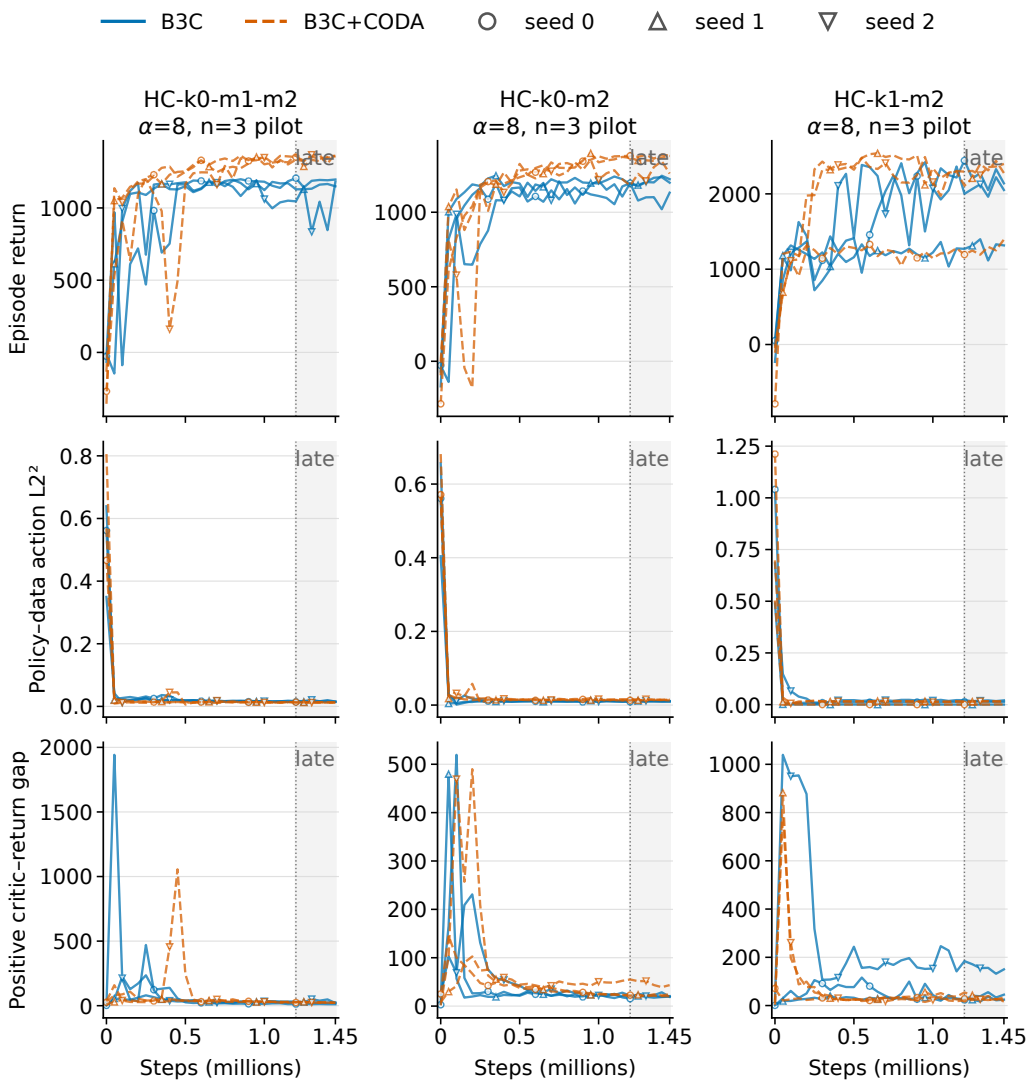


Figure 10: Individual-seed diagnostics for the high-α pilot, with $\alpha = 8$ fixed for both methods. Columns correspond to the three pilot settings; rows show evaluation return, policy–data action L2-squared, and the positive critic–return gap. Each curve is one run, not a seed average. Color and line style distinguish B3C from B3C+CODA; marker shape identifies paired seeds 0, 1, and 2. All 30 scheduled checkpoints are shown, and the shaded interval marks the predefined late window, 1.20–1.45 million updates. These trajectories display seed-level variability; checkpoints are not independent replications.

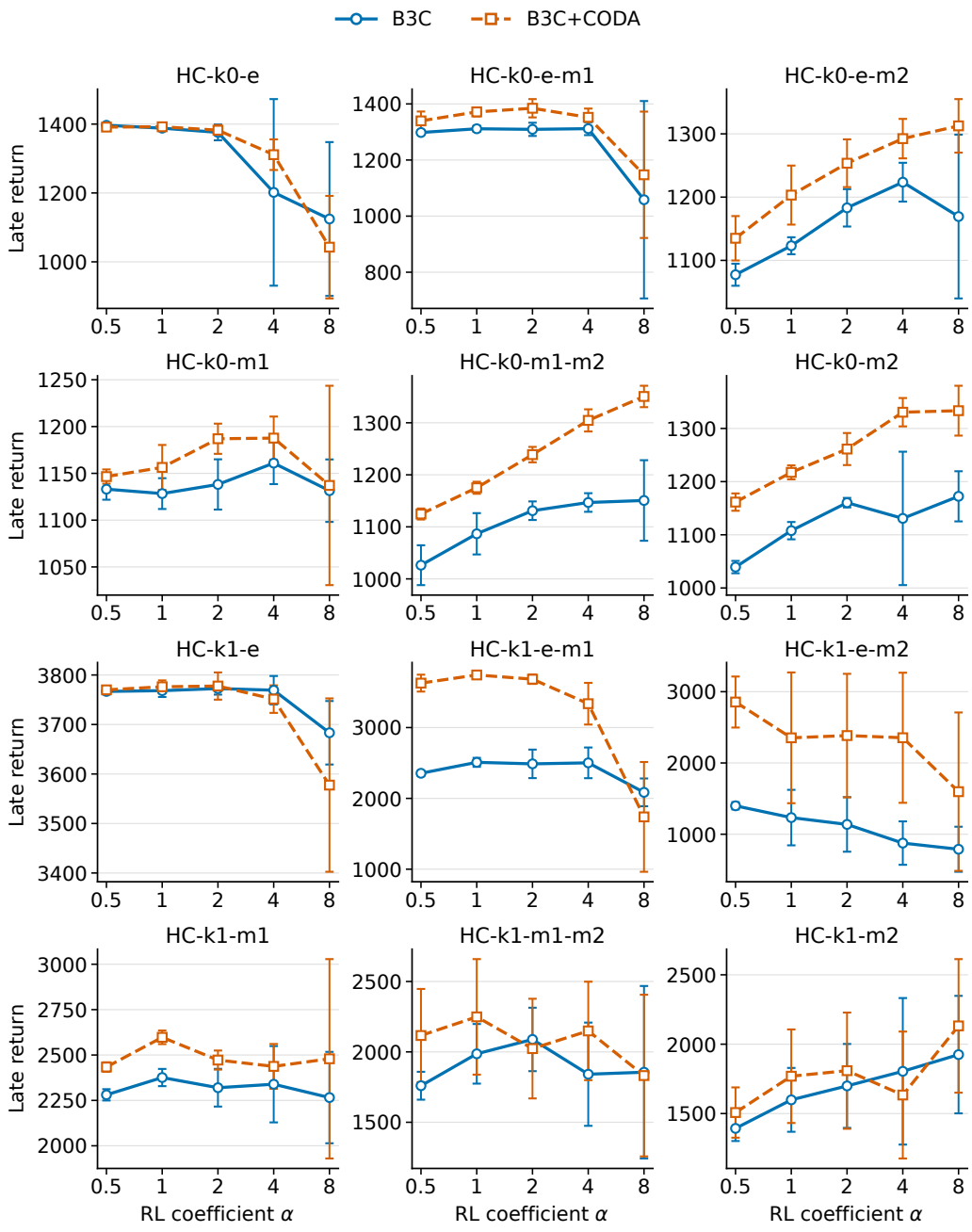


Figure 11: All twelve PO HalfCheetah return–α comparisons from the five-seed grid.

G Complete Regularization Curves

The full 21-setting B3C/CODA grid is shown in addition to the representative main-text panels. Each $\alpha \in \{0.5, 1, 2, 4, 8\}$ is represented by its five-seed late mean and SD, without smoothing across α. A logarithmic base-two horizontal axis gives equal spacing to the tested coefficients. The appropriate comparison is between methods at the same α. An advantage over a range of values and sensitivity relative to each method's own optimum are different properties.

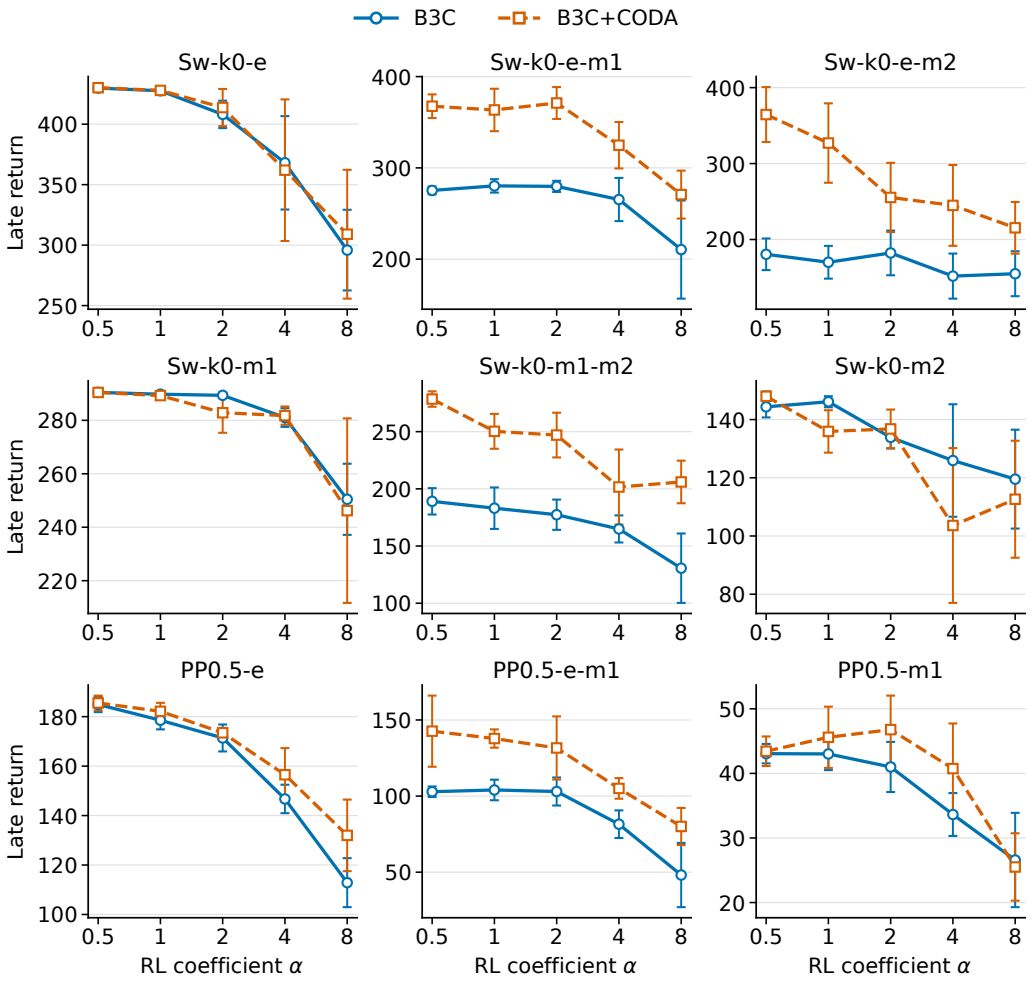


Figure 12: All Swimmer and Predator–Prey return–α comparisons from the five-seed grid.

H DELAYED COMMUNICATION

The delayed model receives other agents' messages one timestep after they are generated. Both training and evaluation use this information pattern. The implementation uses message dimension eight and four consecutive training samples. We evaluate five settings, five values of $\alpha \in \{0.5, 1, 2, 4, 8\}$, and five seeds, giving 125 delayed-model runs. The remaining backbone settings follow the no-delay study. For each condition, α maximizes the five-seed mean return over the final 20% of evaluations. Selection and reporting use the same seeds, as in the main performance grid.

Table 4 reports both conditions' returns in the main text. Table 13 gives the selected coefficients and relative changes. The models are learned separately, so this experiment evaluates learning under a delayed information pattern rather than adding delay to a fixed no-delay policy. It also does not isolate delay from the history processing used by the delayed implementation.

Table 13: Selected RL coefficients for the delayed-communication comparison. Relative changes use no-delay `CODA` as the reference.

Setting	No-delay α	Delayed α	Return change
HC-k0-e	1	0.5	-0.24%
HC-k1-e-m1	1	1	-2.05%
HC-k1-e-m2	0.5	4	-2.25%
Sw-k0-e-m2	0.5	4	-50.62%
PP0.5-e-m1	0.5	4	-10.09%

The delayed model selects a different α in four settings. On HC-k1-e-m2, its mean return is 1843.2 at $\alpha = 0.5$ and 2790.9 at $\alpha = 4$, showing why each communication condition is evaluated with its own search. The three HalfCheetah settings retain similar mean returns under delay, whereas the decrease on Sw-k0-e-m2 is larger.

I ALTERNATIVE COMMUNICATION ARCHITECTURES

We also evaluate RIAL (Foerster et al., 2016) and IC3Net (Singh et al., 2019) as alternative communication architectures in partially observable HalfCheetah. Table 14 compares them with B3C and the attention-based CODA model.

Table 14: Alternative communication architectures on partially observable HalfCheetah.

Setting	B3C	RIAL	IC3Net	Attention CODA
HC-k0-e	1396.4 ± 4.3	1387.6 ± 20.2	1376.3 ± 27.0	1392.2 ± 7.6
HC-k0-m1	1160.9 ± 22.4	1194.8 ± 24.0	1183.8 ± 22.2	1187.7 ± 23.0
HC-k0-m2	1172.4 ± 47.2	1285.0 ± 167.2	1195.1 ± 105.6	1333.4 ± 46.9
HC-k0-e-m1	1311.8 ± 23.0	1145.6 ± 19.5	1282.8 ± 93.3	1384.6 ± 32.6
HC-k0-e-m2	1223.6 ± 30.7	1075.5 ± 2.9	1151.8 ± 37.7	1312.6 ± 42.3
HC-k0-m1-m2	1150.7 ± 77.4	1027.3 ± 11.0	1003.7 ± 19.3	1350.6 ± 20.6
HC-k1-e	3772.6 ± 12.2	3744.0 ± 20.2	3764.3 ± 4.5	3777.7 ± 27.5
HC-k1-m1	2376.0 ± 47.4	2388.2 ± 18.1	2441.1 ± 45.1	2597.4 ± 37.4
HC-k1-m2	1924.8 ± 423.5	2021.2 ± 42.0	2141.5 ± 220.3	2131.8 ± 481.4
HC-k1-e-m1	2509.1 ± 63.4	3546.7 ± 90.1	3695.6 ± 77.1	3742.6 ± 35.8
HC-k1-e-m2	1398.8 ± 51.0	2688.1 ± 61.6	2515.9 ± 441.5	2855.0 ± 358.0
HC-k1-m1-m2	2088.5 ± 225.4	2242.1 ± 112.3	2112.6 ± 152.2	2248.8 ± 410.3