ICMAPE: In-Context Multiagent Pure Exploration

Xinyi Hu [1] **Alessio Russo** [1] **Aldo Pacchiano** [1 2]

Abstract

In some multi-agent systems, the primary objective is not reward maximization but efficient information acquisition to identify unknown properties of the environment as done in active sequential hypothesis testing (ASHT) problems. However, the ASHT literature tends to focus on finite single-agent problems with well-specified models, while there is currently a gap for practical multi-agent methods that can perform active sequential testing. In this work, we fill this gap and propose a Bayesian learning-based framework for decentralized multi-agent pure-exploration driven by inference objectives. Our method, ICMAPE, jointly learns a centralized neural inference network that estimates a posterior distribution over hypotheses from global trajectory data, and decentralized policies trained through reinforcement learning to select actions based only on local observation histories. Instead of relying on environment rewards, we use inference confidence as the training signal, encouraging agents to collect data that reduces uncertainty and enabling them to stop exploration once the inferred hypothesis reaches the target confidence level. On two synthetic benchmarks and a real-world Maryland nitrate concentration monitoring task, ICMAPE-TD3 achieves target accuracy with fewer exploration steps.

1. Introduction

Several existing multi-agent reinforcement learning (MARL) algorithms are designed to maximize cumulative reward (Tan, 1993; Matignon et al., 2012; Son et al., 2019; Lowe et al., 2017; Foerster et al., 2018; Iqbal & Sha, 2019; Ackermann et al., 2019; Sheikh & Bölöni, 2020; Zhang & Guo, 2025; Wei et al., 2025). However, many multi-agent systems have objectives that go beyond reward maximization. In some cases, the goal is to collaboratively collect informative data in order to identify unknown properties of the environment while minimizing data collection cost, under local observation constraints. For example, in distributed sensor localization, multiple sensors may need to combine local measurements to infer an unknown target or source location (Patwari et al., 2005; Mao et al., 2007; Paul & Sato, 2017). In these scenarios, coordination is necessary for information gain rather than reward accumulation. Such a perspective is closely related to active sequential hypothesis testing and pure exploration, where actions are selected adaptively to identify the true hypotheses with minimal sampling cost (Naghshvar & Javidi, 2013; Kaufmann et al., 2016; Garivier & Kaufmann, 2016; Russo, 2016; Russo et al., 2025; Vannella et al., 2024). Related multi-agent and federated variants extend pure exploration beyond the single-agent setting, but often retain centralized action selection or focus primarily on communication constraints rather than decentralized data collection (Vannella et al., 2023; Mitra et al., 2021; Kota et al., 2023; Chen et al., 2024).

Decentralized multi-agent pure exploration problems introduce several key challenges. First, each agent must make decisions from only its local trajectory, without observing the full joint exploration history. Second, because the hypothesis is identified from the collective data, the decentralized policies must induce coordinated exploration patterns that are informative at the global level rather than redundant. Finally, agents must learn when enough evidence has been collected to stop, since continuing to sample after the hypothesis is already identifiable only increases data-collection cost.

Standard cooperative MARL approaches under centralized training with decentralized execution (CTDE) , such as VDN (Sunehag et al., 2017), QMIX (Rashid et al., 2020), MADDPG (Lowe et al., 2017), COMA (Foerster et al., 2018), MATD3 (Ackermann et al., 2019), and MAPPO (Zhang & Guo, 2025), are typically formulated for maximizing cumulative environment reward. As a result, they do not directly optimize information gain or inference confidence. If those methods are applied to ASHT or pure-exploration problems using the original environment reward, agents may learn high-reward behaviors that are not informative enough to identify the underlying hypothesis.

[1]Faculty of Computing & Data Sciences, Boston University, Boston, MA, USA [2]Broad Institute of MIT and Harvard, Cambridge, MA, USA. Correspondence to: Xinyi Hu <xhu07@bu.edu>.

ICML 2026 Workshop on Hypothesis Testing, Seoul, South Korea, 2026. Copyright 2026 by the author(s).

Alternative strategies, such as pure exploration methods, directly optimize data collection for the target inference task. However, many such methods are tailored to specific bandit models, limiting their transfer to new decentralized multi-agent inference problems and are not well-suited to this setting (Kaufmann et al., 2016; Garivier & Kaufmann, 2016).

We propose *In-Context Multi-Agent Pure Exploration* (ICMAPE), a learning-based framework for decentralized multi-agent exploration inspired by In-Context Pure Exploration (ICPE) (Russo et al., 2026b). ICMAPE jointly learns a centralized inference network $\mathcal{I}$, which maps global trajectory data to a posterior over hypotheses, and decentralized policies that collect data from local observation histories. Here, a hypothesis denotes a possible answer to the inference task, such as a target location or latent environment parameter. The key challenge is that inference depends on the joint trajectory, while data collection is decentralized. ICMAPE addresses this by using global inference confidence and a centralized critic to train local policies, leading to accurate identification with shorter stopping times across our tasks.

2. Related Work

Active Sequential Testing. Our work is closely related to active sequential hypothesis testing (Naghshvar & Javidi, 2013) and fixed-confidence pure exploration (Kaufmann et al., 2016; Garivier & Kaufmann, 2016), both of which adaptively collect data to identify a target hypothesis while minimizing expected stopping time subject to a correctness constraint. Classical approaches derive sampling and stopping rules for specific statistical models, including Chernoff-style active hypothesis testing, Track-and-Stop, and Top-Two Sampling methods (Naghshvar & Javidi, 2013; Kaufmann et al., 2016; Garivier & Kaufmann, 2016; Russo, 2016; Vannella et al., 2024; Degenne et al., 2019; Jourdan et al., 2022; Wang et al., 2021; Russo & Pacchiano, 2025). Recent work on In-Context Pure Exploration (ICPE) treats exploration as a learnable object across problem instances (Russo et al., 2025).

Closest to our setting, Szostak and Cohen (Szostak & Cohen, 2024) study decentralized active hypothesis testing using deep multi-agent reinforcement learning, where agents select both sampling actions and a stopping action, and rewards are derived from Bayesian risk. However, their method requires posterior beliefs to be computed from known observation models, which may not be available in more general environments. Moreover, they do not theoretically justify the stopping action or the learning objective. In contrast, we extend the learn-to-explore perspective to decentralized multi-agent settings by jointly learning decentralized data-collection policies π and a centralized inference rule $\mathcal{I}$, using a generalizable inference-based reward and providing theoretical guarantees for the the optimal inference rule, stopping action, and the derivation of the learning objective from the fixed-confidence formulation; see Proposition A.1, Lemma A.3, and Proposition A.3.

Federated and Distributed Pure Exploration. Recent works have extended best-arm identification (BAI) to multi-agent settings. In federated BAI, multiple agents explore bandit arms and communicate with a central server that aggregates information and identifies the arm with the highest mean reward across agents, while minimizing data exposure (Mitra et al., 2021; Kota et al., 2023; Chen et al., 2024). Another related line studies multi-agent best arm identification with centralized action selection under structured reward models such as factor graphs (Vannella et al., 2023). These approaches derive instance-dependent lower bounds and design algorithms based on Track-and-Stop principles and likelihood-based stopping rules (Garivier & Kaufmann, 2016). However, these formulations assume either centralized coordination or centralized action selection, and focus on identifying an optimal joint action. In contrast, our setting requires decentralized action selection from local observation histories, while the agents jointly collect data to solve a broader inference task.

Cooperative Multi-Agent Reinforcement Learning. Learning decentralized policies in multi-agent settings is challenging because each agent's learning process changes the environment faced by the others, often leading to instability in independent learning (Matignon et al., 2012). To address this, we adopt the CTDE paradigm, where agents are trained in a centralized manner but execute using only local observations. Within CTDE, two main classes of approaches have been proposed. Value factorization methods, such as VDN (Sunehag et al., 2017), QMIX (Rashid et al., 2020), and QTRAN (Son et al., 2019), decompose a joint value function into per-agent components. Policy gradient methods, including MADDPG (Lowe et al., 2017), COMA (Foerster et al., 2018), MAAC (Iqbal & Sha, 2019), and MATD3 (Ackermann et al., 2019), instead learn decentralized policies using centralized critics. While these approaches have shown success in cooperative control tasks, they are primarily designed for reward maximization and do not directly address pure exploration or inference objectives.

3. Background

3.1. Problem Formulation

We study a decentralized multi-agent sequential decision problem aimed at identifying an unknown property of the environment. Agents interact with the environment while

having access only to their local observations to collect data. The objective is to efficiently gather informative data to enable accurate inference under a fixed-confidence requirement. We now formalize the setting.

Environment Modeling. We consider a system with N agents interacting with a common environment drawn from a model class $\mathcal{M}$, under a distribution $\mathcal{P} \in \Delta(\mathcal{M})$. Each environment is modeled as a tuple $M = (\mathcal{X}, \mathcal{A}, P, \rho)$, where $\mathcal{X} = \bigcup_{i=1}^{N} \mathcal{X}^i$ is the union of agent-specific observation spaces, and $\mathcal{A} = \bigcup_{i=1}^{N} \mathcal{A}^i$ is the union of agent-specific action sets. The transition function is $P = (P_t)_{t\in\mathbb{N}}$, where $P_t : (\mathcal{X} \times \mathcal{A})^t \to \Delta(\mathcal{X})$.

Agents and Decentralized Policies. Agents act simultaneously at each time step. Each agent $i \in \{1, \dots, N\}$ receives a local observation $x_t^i \in \mathcal{X}^i$, selects an action $a_t^i \in \mathcal{A}^i$, and then receives the next local observation x_{t+1}^i. Denote the joint action by $a_t = (a_t^1, \dots, a_t^N)$, and let $\mathcal{D}_t$ denote the full joint trajectory. The local trajectory available to agent i is $\mathcal{D}_t^i = (x_1^i, a_1^i, \dots, a_{t-1}^i, x_t^i)$. We denote the joint observation by $x_t = (x_t^1, \dots, x_t^N)$, and the full joint trajectory by $\mathcal{D}_t = \{\mathcal{D}_t^i\}_{i=1}^N$.

We restrict attention to decentralized policies. Each agent follows a policy $\pi^i = (\pi_t^i)_{t\in\mathbb{N}}$ satisfying $\pi_t^i : \mathcal{D}_t^i \to \Delta(\mathcal{A}^i)$. Thus, agent i samples actions according to $a_t^i \sim \pi_t^i(\cdot \mid \mathcal{D}_t^i)$, using only its own local trajectory. The joint policy is $\pi = (\pi^1, \dots, \pi^N)$, and the induced joint action distribution factorizes as $\pi_t(a_t \mid \mathcal{D}_t) = \prod_{i=1}^N \pi_t^i(a_t^i \mid \mathcal{D}_t^i)$.

Hypothesis. Within the collective observation space $\mathcal{X}$, we consider a predefined task-specific hypothesis class $\mathcal{H}$ and restrict our attention to the finite case (extensions to continuous classes are possible (Russo et al., 2026a)). For any environment $M \in \mathcal{M}$, there exists a ground-truth hypothesis $H_M^* \in \mathcal{H}$. We drop the subscript M and write H^* when the dependence is clear from context. The objective is to ensure that the estimated hypothesis $\hat{H} \in \mathcal{H}$ matches the ground-truth hypothesis H^*.

Objective. Given a trajectory of observations collected by the agents, the goal is to accurately identify the ground-truth hypothesis. To formalize this, we introduce an inference function that maps observed data to a distribution over hypotheses. The inference function operates on the full joint trajectory $\mathcal{D}_t = \{\mathcal{D}_t^i\}_{i=1}^N$ and is defined as $\mathcal{I} : \mathcal{D}_t \mapsto \Delta(\mathcal{H})$. Given a target error probability $\delta \in (0, 1)$, the decentralized data-collection policies π and the inference rule $\mathcal{I}$ induce a stopping time τ, which denotes the number of joint interaction rounds before termination. The objective is to minimize the expected stopping time while ensuring that the final prediction is correct with probability at least $1 - \delta$:

$$\min_{\pi,\tau,\mathcal{I}} \mathbb{E}^\pi[\tau] \quad \textbf{s.t.} \quad \mathbb{P}^\pi\big(\mathcal{I}(\mathcal{D}_\tau) = H^\star\big) \geq 1 - \delta. \tag{1}$$

Intuitively, we seek the cheapest data-collection strategy that allows a centralized inference rule to identify the true hypothesis with confidence $1 - \delta$. Here, $\mathbb{P}^\pi(\cdot)$ denotes the probability measure over trajectories induced by first sampling an environment $M \sim \mathcal{P}$ from the model class $\mathcal{M}$ and then executing policy π.

4. Method

Our method begins by reformulating the constrained identification problem through a Lagrangian relaxation. We then optimize over the inference rule, showing that the optimal rule selects the hypothesis with maximal posterior confidence. This yields a reward-based sequential decision problem with an auxiliary stopping action, which we solve using an actor–critic reinforcement learning framework under a CTDE paradigm.

4.1. Lagrangian Relaxation and RL Reformulation

We now show how the constrained optimization problem in Eq. 1 can be reformulated as a reinforcement learning problem. The key idea is to first relax the fixed-confidence constraint using a Lagrangian multiplier, then optimize over the inference rule, and finally express the resulting objective as a reward-maximization problem with a stopping action.

Introducing a multiplier $\lambda \geq 0$, the Lagrangian relaxation of Eq. 1 is given by

$$\inf_{\lambda\geq 0} \sup_{\pi,\mathcal{I}} \Big[-\mathbb{E}_\pi[\tau] + \lambda\Big(\mathbb{P}_\pi\big(\mathcal{I}(\mathcal{D}_\tau) = H^\star\big) - 1 + \delta\Big)\Big]. \tag{2}$$

For any fixed policy π, we can optimize over the inference rule. At any time $t \in \mathbb{N}$, given the observed trajectory $\mathcal{D}_t$, the optimal inference rule selects the hypothesis with the largest posterior probability (Proposition A.1):

$$\mathcal{I}_t(\mathcal{D}_t) \in \arg\max_{H\in\mathcal{H}} \mathbb{P}_t(H^* = H \mid \mathcal{D}_t),$$

where $\mathbb{P}_t(\cdot \mid \mathcal{D}_t)$ denotes the posterior distribution over hypotheses. This induces the posterior confidence

$$r_t(\mathcal{D}_t) := \max_{H\in\mathcal{H}} \mathbb{P}_t(H^* = H \mid \mathcal{D}_t)$$

, which represents the probability of correct identification under the optimal inference rule. Substituting it into the Eq. 2 yields

$$\inf_{\lambda\geq 0} \sup_{\pi} \mathbb{E}^\pi\big[-\tau + \lambda\big(r_\tau(\mathcal{D}_\tau) - 1 + \delta\big)\big].$$

To express this objective as a sequential decision problem, we augment each agent's action space with an auxiliary stopping action a_{stop}. At each time step, agents may either continue collecting data or trigger termination. We assume that stopping is shared across agents, meaning if any agent selects the stopping action, the data collection process terminates and the stopping signal is observed by all agents. This induces a common stopping time, formalized below.

Assumption 4.1 [Common stopping time]. *There exists a universal stopping time τ such that for every agent i, τ is a stopping time with respect to the local filtration $\mathcal{F}_t^i = \sigma(\mathcal{D}_t^i)$; i.e., for all $t \geq 1$, the event $\{\tau \leq t\} \in \mathcal{F}_t^i$. Equivalently, $\{\tau \leq t\} \in \bigcap_{i=1}^N \mathcal{F}_t^i$.*

Under this formulation, continuing exploration incurs a unit cost, while stopping receives a terminal reward based on the posterior confidence of the inferred hypothesis. We define the reward function as

$$r_{t,\lambda}(\mathcal{D}_t, a_t) = -\mathbf{1}\{a_t \neq a_{\text{stop}}\} + \lambda\big(r_t(\mathcal{D}_t) - 1 + \delta\big)\mathbf{1}\{a_t = a_{\text{stop}}\}.$$

This reward function converts the original fixed-confidence problem into a reward-maximization problem. The corresponding action-value function satisfies the Bellman recursion

$Q_{t,\lambda}(\mathcal{D}_t, a) = r_{t,\lambda}(\mathcal{D}_t, a) + \Big(\mathbf{1}\{a \neq a_{\text{stop}}\} \cdot \mathbb{E}_{x_{t+1}\sim P_t(\cdot|\mathcal{D}_t,a)}\Big[\max_{a'} Q_{t+1,\lambda}\big((\mathcal{D}_t, a, x_{t+1}), a'\big)\Big]\Big)$. The expectation is taken over the next observation induced by the environment transition kernel given the current trajectory and action. Details of the dual formulation and the resulting reward maximization problem are provided in Appendix A.3.

4.2. Inference Network

We introduce an inference network that extracts task-relevant information from the global trajectory data collected by all agents. Let $H^\star$ denote the true underlying hypothesis of the environment $M \sim \mathcal{P}$. For example, in a source localization task, $H^\star$ may be the hidden location of the signal source.

Given the trajectory up to time t, denoted by $\mathcal{D}_t$, the inference network $\mathcal{I}_\psi$, parameterized by ψ, outputs a posterior distribution over hypotheses, $p_\psi(H \mid \mathcal{D}_t)$. We obtain a point estimate via maximum a posteriori (MAP) inference, given by $\hat{H}_t = \arg\max_{H\in\mathcal{H}} p_\psi(H \mid \mathcal{D}_t)$. The inference network is trained using the true hypothesis $H^\star$ sampled with each environment instance by minimizing the negative log-likelihood loss

$$\mathcal{L}_{\text{inf}} = -\log p_\psi(H^\star \mid \mathcal{D}_t). \tag{3}$$

Importantly, the reward used for policy learning is constructed based on the confidence of this posterior. Specifically, rather than relying on an external reward signal, we define the reward as

$$r_t = p_\psi(\hat{H}_t \mid \mathcal{D}_t),$$

the posterior confidence assigned to the inferred hypothesis. This reward is then used to train the critic, which estimates value functions under the induced reward signal, and the policy (actor network) is optimized with respect to the critic, and thus indirectly encouraged to collect trajectories that improve inference of the underlying environment.

4.3. Critics

The critic network uses the reward signal defined in Section 4.2 to evaluate the quality of actions. We build on MATD3 (Ackermann et al., 2019), which extends MADDPG (Lowe et al., 2017) with the twin-critic architecture of TD3 (Fujimoto et al., 2018). During training, the centralized critic conditions on the global trajectory and joint action, while the actor remains decentralized and depends only on the agent's local trajectory. The two Q-functions, Q_{ϕ_1} and Q_{ϕ_2}, are used through a minimum operator in the Bellman target to reduce overestimation bias. Since multiplication by the positive constant λ does not change the optimal policy for fixed λ, we parameterize the tradeoff by a learnable per-step cost $\zeta \approx 1/\lambda$. Therefore, continuing exploration incurs cost of $-\zeta$, while stopping receives the terminal confidence reward.

For each agent i, the critic estimates an agent-specific value $Q^i_{\phi_j}(\mathcal{D}_t, a_t^i, a_t^{-i})$ for $j \in \{1, 2\}$, from a combined representation of the global trajectory and the joint action. The global trajectory $\mathcal{D}_t$ is encoded using the sequence encoder described in Section B.7, and the joint action is encoded by concatenating one-hot representations of all agents' actions. Here, a_t^i denotes agent i's own action and a_t^{-i} denotes the actions of the other agents.

We define the immediate cost as

$$c_t = \begin{cases} -1, & \text{if no agent stops at time } t, \\ r_t(\mathcal{D}_t) - 1 + \delta, & \text{otherwise,} \end{cases} \tag{4}$$

where $r_t(\mathcal{D}_t)$ is the posterior confidence of the inferred hypothesis. Let $s_t = \mathbf{1}\{\exists i,\ a_t^i = a_{\text{stop}}\}$ denote whether any agent stops at time t. Then the TD3 target for agent i is

$$y_t^i = \zeta c_t + \gamma(1 - s_t) \min_{\ell\in\{1,2\}} Q^i_{\phi_\ell^{\text{tar}}}\big(\mathcal{D}_{t+1}, \tilde{a}^{i,\text{tgt}}_{t+1}, \tilde{a}^{-i,\text{tgt}}_{t+1}\big),$$

where γ is the discount factor, $Q_{\phi_1^{\text{tar}}}$ and $Q_{\phi_2^{\text{tar}}}$ are target critics, and $\tilde{a}^{\text{tgt}}_{t+1}$ is obtained from the target actor. The scalar $\zeta \geq 0$ is a learned multiplier that balances exploration

cost and terminal confidence reward. Given the batch-level posterior confidence $\widehat{r}_t$, we update it using the dual-variable rule $e_\zeta = \widehat{r}_t - (1-\delta)$ and $\zeta \leftarrow \max\{0, \zeta + \eta e_\zeta\}$, where η is the learning rate. We initialize $\zeta = 0$, so the exploration cost is initially inactive and increases only when the posterior confidence exceeds the target level $1-\delta$.

Finally, the critic parameters are updated by minimizing the temporal-difference loss

$$\mathcal{L}_{\text{critic}} = \mathbb{E}\left[\sum_{i=1}^{N}\sum_{\ell=1}^{2}\left(Q^i_{\phi_\ell}(\mathcal{D}_t, a^i_t, a^{-i}_t) - y^i_t\right)^2\right]. \quad (5)$$

4.4. Actors

For each agent i, the actor parameterizes a decentralized policy on the local trajectory $\mathcal{D}^i_t$. The actor maps a learned representation of $\mathcal{D}^i_t$ to unnormalized action scores $g^i_t = (g^i_t(a))_{a\in\mathcal{A}^i}$. At execution time, we use the deterministic policy $a^i_t = \boldsymbol{\mu}_{\theta_i}(\mathcal{D}^i_t) = \arg\max_{a\in\mathcal{A}^i} g^i_t(a)$. During actor training, however, we use a Gumbel–Softmax relaxation to obtain differentiable action representations $\widetilde{a}^i_t$, allowing gradients to pass through the discrete action-selection process.

The decentralized actors are trained to maximize the centralized critic value under the relaxed joint action $\widetilde{a}_t = (\widetilde{a}^1_t, \ldots, \widetilde{a}^N_t)$. We define the actor objective

$$J(\theta) = \mathbb{E}\left[\sum_{i=1}^{N} Q^i_{\phi_1}\left(\mathcal{D}_t, \widetilde{a}^i_t, \widetilde{a}^{-i}_t\right)\right], \quad (6)$$

and optimize the actors by minimizing $\mathcal{L}_{\text{actor}}(\theta) = -J(\theta)$. Following TD3, the actor is updated less frequently than the critic to improve training stability; in our implementation, we use an update frequency of 5.

The pseudocode for training and evaluation (execution) is provided in Algorithms 1 and 2. Unlike the centralized critic, which uses global information during training to evaluate joint actions, the actors are executed in a decentralized manner. At inference time, each agent collects data using its own actor network, which has access only to the agent's local trajectory. After data collection terminates, the inference network aggregates the global trajectory and predicts the underlying hypothesis. Furthermore, implementation details for network architectures are provided in Appendix B.5.

5. Experiments

We evaluate ICMAPE-TD3 in two synthetic benchmark environments designed to isolate key challenges in decentralized information acquisition. First, we study bandit capacity identification, where agents must infer how many agents each arm can support before becoming overloaded. This task requires coordinated joint actions to reveal meaningful information about the hypothesis and is motivated by shared resource systems such as communication channels with unknown user capacity. We consider two bandit settings: a 3-arm-5-player setting (A3P5) and a 5-arm-3-player setting (A5P3).

Second, we study signal source localization with distractors, motivated by distributed sensor localization applications such as environmental monitoring, bush fire surveillance, and precision agriculture, where agents coordinate exploration to identify the true signal source (Mao et al., 2007; Patwari et al., 2005). Third, we apply ICMAPE-TD3 to a real-world environmental monitoring task using Maryland nitrate concentration data. Environment details are provided in Sections 5.1, 5.2, and 5.3.

5.1. Multi-armed Bandits Capacity Identification

We consider a setting with K arms and N players. Inspired by multiplayer multi-armed bandit models with non-unit arm capacities (Wang et al., 2022a,b; Hu & Pacchiano, 2026), we study the problem of identifying the capacity of each arm $k \in [K]$ with confidence level $1-\delta$, where $\delta \in [0, 1)$.

Each arm k has an unknown capacity $C_k \in \{1, \ldots, N\}$, which is sampled independently from a uniform distribution over $\{1, \ldots, N\}$, defined as the maximum number of agents that can simultaneously select arm k without violating its constraint. At each time step t, each agent i chooses either an arm to pull or the stopping action, so that

$$a^i_t \in [K] \cup \{a_{\text{stop}}\}.$$

If $a^i_t = k$, then agent i pulls arm k at time t; if $a^i_t = a_{\text{stop}}$, the agent triggers termination and no further exploration is performed. Let $n^k_t = \sum_{i=1}^{N} \mathbb{I}\{a^i_t = k\}$ denote the number of agents selecting arm k at time t. When $n^k_t > C_k$, the arm is said to be overloaded, and the resulting observations are zero environment rewards on that arm. Otherwise, each agent i pulling arm k receives an outcome $x^i_t \sim \text{Bernoulli}(\mu_k)$, where $\mu_k \in (0, 1]$ is the mean reward of arm k. To ensure that the capacity identification problem remains tractable, we assume the mean reward is first sampled as $\mu_k \sim \text{Unif}[0.3, 1]$ for all arms. This assumption avoids the setting in which an impractically large number of samples would be required to reliably estimate the capacity.

In our framework, the inference network operates on the global trajectory and observation $\mathcal{D}_t$ aggregated across all agents to produce an estimate of the underlying hypothesis. In contrast, each agent selects actions based only on its local information $\mathcal{D}^i_t$, without direct access to the global information. In this experiment, the local trajectory $\mathcal{D}^i_t = \{(a^i_s, x^i_s)\}^t_{s=1}$, while the global trajectory is given by $\mathcal{D}_t = \{(a_s, x_s)\}^t_{s=1}$, where $a_s = (a^1_s, \ldots, a^N_s)$ denotes the joint

action and $x_s = (x_s^1, \ldots, x_s^N)$ are the corresponding bandit outcomes.

Objective and Inference Reward The goal is to identify the true capacity vector $C^\star = (C_1^\star, \ldots, C_K^\star)$ with confidence level $1-\delta$. The hypothesis space is defined as $\mathcal{H} = \{1, \ldots, N\}^K$, where each hypothesis corresponds to a possible capacity configuration over the K arms. We further define the reward in terms of the probability of correctly identifying the true capacity vector. Let $\hat{C}_t = \arg\max_{C \in \mathcal{H}} p_\psi(C \mid \mathcal{D}_t)$ denote the inferred capacity vector. The reward is given by $r_t = p_\psi(\hat{C}_t \mid \mathcal{D}_t)$, which corresponds to the posterior confidence assigned to the inferred capacity.

5.2. Signal Source Localization with Distractors

In the second environment, we model a cooperative signal-localization problem in which heterogeneous agents seek to identify an unknown signal source through decentralized exploration. We consider a discrete grid world of size $n \times n$, with $n = 25$, and $N = 3$ agents. The objective is to identify the true signal source location $s^\star \in [n] \times [n]$ with confidence level $1-\delta$, where $\delta \in [0, 1)$.

Hypothesis. The environment contains one true signal source $s^\star = (r^\star, c^\star) \in [n] \times [n]$ and one distractor source $s' = (r', c') \in [n] \times [n]$. The hypothesis space is $\mathcal{H} = [n] \times [n]$, where each hypothesis corresponds to a possible true source location. Both sources generate higher signal intensity than nearby cells. The true source has peak intensity 1, while the distractor has peak intensity ξ, where $\xi \sim \mathrm{Unif}[0, 0.7]$. To ensure that the two sources are spatially distinguishable, we impose $d_{\mathrm{Man}}(s^\star, s') \geq 10$.

Actions and Observations. At each time step t, each agent i is located at position $p_t^i \in [n] \times [n]$ and chooses either a movement action or the stopping action, so that

$$a_t^i \in \{\texttt{up}, \texttt{down}, \texttt{left}, \texttt{right}, \texttt{stay}\} \cup \{a_{\mathrm{stop}}\}.$$

The movement actions update the agent's position by moving to an adjacent grid cell with step size Δ_{move}, with boundary conditions preventing movement outside the grid. The action `stay` keeps the agent at its current location.

Each agent observes only its own sensing modality and does not directly observe the measurements of other agents. However, agents can observe the positions of the other agents. At time t, each agent observes an $m \times m$ local window centered at its current position p_t^i, with boundary clipping at the grid edges. For agents 1 and 2, the observation at any queried cell u follows

$$Y^{(i)}(u) \sim \mathrm{Bernoulli}(q_i) \cdot \mathbb{I}\{u = s^\star\}$$

, where $s^\star$ denotes the target cell and q_i is the detection probability of agent i.Agent 3 follows a different observation model, described below.

Agent 3 observes a noisy version of the latent signal field. For each cell $u \in [n] \times [n]$, let $d_{\min}(u) = \min\{d_{\mathrm{Man}}(u, s^\star), d_{\mathrm{Man}}(u, s')\}$ and define

$$s(u) = I(u) \exp(-\lambda d_{\min}(u)),$$

where $I(u) = 1$ for cells closer to $s^\star$ and $I(u) = \xi$ otherwise, and decay rate $\lambda > 0$. Agent 3 observes

$$Y^{(3)}(u) = \mathrm{clip}_{[0,1]}\big(s(u) + \epsilon(u)\big), \text{ where } \epsilon(u) \sim \mathcal{N}(0, \sigma^2).$$

Example noisy observations are shown in Figure 1. Environment parameters are listed in Appendix Table 1.

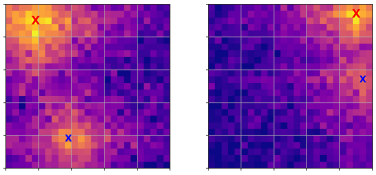


Figure 1. Example of Agent 3's signal field $Y^{(3)}$. Red $\times$: true source; blue $\times$: distractor.

5.3. Maryland Nitrate Concentration Monitoring

We evaluate ICMAPE-TD3 on a real-world environmental monitoring task using Maryland nitrate concentration data spanning 2001–2023, with training on 2001–2018 and evaluation on 2019–2023. We obtained Maryland nitrate concentration measurements from the Water Quality Portal (WQP) (National Water Quality Monitoring Council, 2024), a public database integrating water-quality observations from USGS, EPA, and state/local monitoring agencies. Elevated nitrate levels contaminate drinking water sources and pose direct health risks, particularly for infants (U.S. Environmental Protection Agency, 2026), while also contributing to nutrient pollution, algal blooms, and oxygen depletion that threaten aquatic ecosystems (U.S. Geological Survey, 2026). Therefore, it is important to develop methods that can quickly identify high-concentration regions and support timely intervention.

For each year, a subset of real monitoring measurements is first used to fit a Gaussian process, producing a spatial concentration field $f : \mathcal{X} \to \mathbb{R}$ over a discretized map $\mathcal{X}$ of the region. The true hypothesis $\theta^\star \in \mathcal{H} = \mathcal{X}$ is defined as the location of the highest concentration, i.e., $\theta^\star = \arg\max_{x \in \mathcal{X}} f(x)$. Full experimental details and results are provided in Appendix B.4.

5.4. Baselines

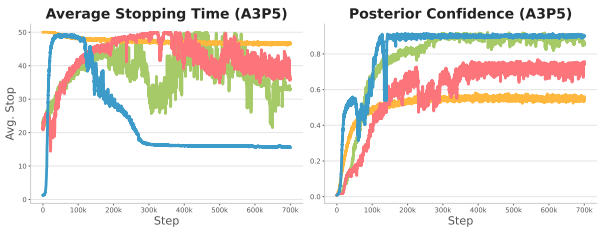
We compare against three baselines used across all experiments: random exploration (Appendix B.2), Independent Q-learning (Appendix B.2), and VDN (Appendix B.2). These baselines span different coordination assumptions and contrast with our policy-gradient actor–critic approach. In addition, we include experiment-specific baselines: environment-reward-guided exploration for the bandit task (Appendix B.2), Thompson Sampling for signal-source localization (Appendix B.2), and GP Prior Map Guided Sampling for the nitrate monitoring task (Appendix B.2).

6. Results

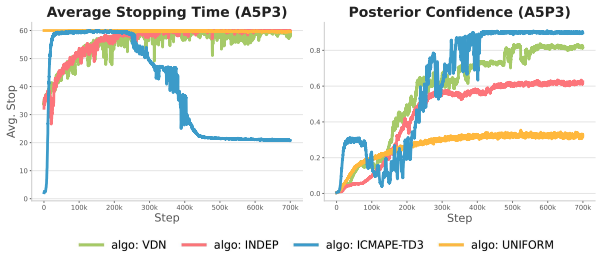
As described in Sections 5.1, 5.2, and 5.3, we evaluate our method, ICMAPE-TD3, on the bandit capacity identification task, the signal source localization task, and a real-world Maryland nitrate concentration monitoring task, along with the baseline methods described in Section B.2.

6.1. Multi-armed Bandit Capacity Identification

Training. The training dynamics for both settings are shown in Figure 2. For each setting, we report the average stopping time and the posterior confidence assigned by the inference network over training steps.



(a) Training dynamics for A3P5.



(b) Training dynamics for A5P3.

Figure 2. Training dynamics in A3P5 and A5P3. For each setting, we report the average stopping time and posterior confidence. Curves are averaged over three random seeds and smoothed with an exponential moving average parameter of 0.2.

We set the confidence level to $\delta = 0.1$, corresponding to a target accuracy of $1 - \delta = 0.9$. In the A3P5 setting, both TD3 and VDN are able to reach the target accuracy early in training. However, once the reward structure begins to favor shorter trajectories more strongly, TD3 adapts more effectively by reducing the average stopping time. This suggests that TD3 learns an effective sampling strategy that reduces stopping time after reaching the target accuracy. In contrast, VDN does not exhibit a similar stable reduction, resulting in higher stopping times despite achieving comparable accuracy.

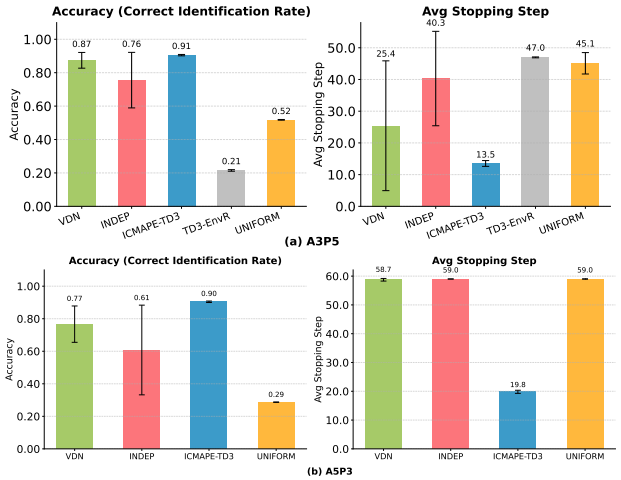


Figure 3. Final performance comparison across methods in the A3P5 and A5P3 settings, averaged over three random seeds. The grey TD3-EnvR bar represents TD3 trained to maximize the environmental reward. This plot shows estimated probability of correct identification and the average stopping time (lower is better). Error bars indicate one standard deviation.

In the A5P3 setting, the capacity identification problem becomes more challenging due to the larger hypothesis space. In this case, only TD3 is able to reliably reach the target accuracy and improve the sampling strategy.

Evaluation. We further evaluate the final performance of each method in terms of estimation accuracy and average stopping time across 10,000 sampled environments over three seeds. The results are summarized in Figure 3. TD3 achieves stable accuracy around $1 - \delta$ while maintaining significantly lower stopping times compared to baselines.

Coordination Strategy. The learned policy follows a highly structured probing pattern. Early joint actions are identical across active environments, first concentrating players on one arm and then reallocating subsets across arms (Table 4 in Appendix). This suggests that agents learn to test capacity constraints through coordinated joint allocations rather than independent random exploration. Once sufficient evidence is collected, one or more agents select the stop action (Table 5 in Appendix).

Robustness Analysis. We evaluate the robustness of ICMAPE-TD3 under different priors over arm means and arm capacities. For the arm-mean prior, we replace the uniform prior with a scaled Beta distribution parameterized by (α, β), where $\alpha, \beta \in \{0.5, 1, 3, 5, 7\}$. When $\alpha > \beta$, the prior places more mass on larger arm means, yielding more informative feedback and average accuracy above 0.95; when $\alpha \leq \beta$, the accuracy ranges from 0.67 to 0.96.

For the arm-capacity prior, we sample capacities from a distribution induced by $p \sim \mathrm{Dir}(\theta \mathbf{1}_K)$, with $\theta \in \{0.1, 0.5, 1, 5, 10\}$. Smaller θ values produce more sparse capacity distributions, while larger values approach a more balanced distribution. ICMAPE-TD3 remains robust across these capacity priors. Figures 9 and 10 in the appendix report the average inference accuracy under different arm-mean and arm-capacity priors.

6.2. Signal Source Localization with Distractors

Training. We also set the error tolerance to $\delta = 0.1$, corresponding to a target accuracy of $1 - \delta = 0.9$. Figure 4 shows the training dynamics for the signal localization task. We plot the average stopping time and the posterior confidence assigned by the inference network over training steps. Similar to the bandit setting, TD3 is the only method that reliably reaches the target accuracy while further improving efficiency by reducing the average stopping time.

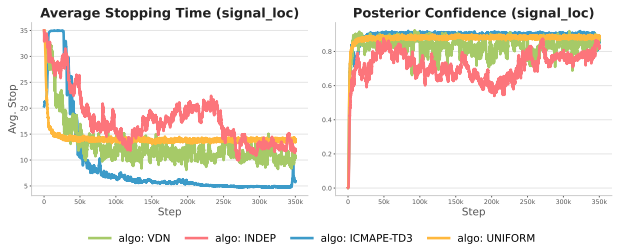


Figure 4. Signal Source Localization Training Dynamics. Curves are averaged over three random seeds and smoothed with an exponential moving average parameter of 0.2.

Evaluation. We evaluate the performance of each method across 10,000 sampled environments using two metrics: identification accuracy and average stopping time. The results are summarized in Figure 5. TD3 achieves accuracy close to the target confidence level $1 - \delta$ while uses fewer samples than the baselines.

Coordination Strategy. In this task, the learned policy exhibits an implicit division of roles. Agents 1 and 2 follow deterministic trajectories across environments, while Agent 3 adapts its actions based on its local observations of the noisy signal field (Table 6 in Appendix). Stopping is also role-specific, with more than one agent stopping simultaneously in fewer than 10% of sampled environments (Table 7 in Appendix).

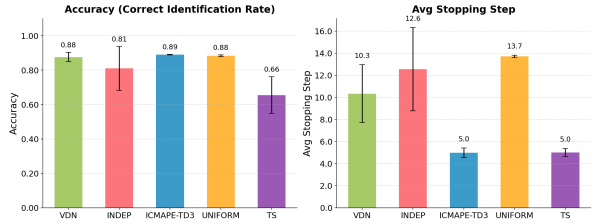


Figure 5. Signal Source Localization Final Performance. Final performance comparison across methods, averaged over three random seeds. This plot shows estimated probability of correct identification and the average stopping time (lower is better). Error bars indicate one standard deviation.

Robustness Analysis. Similar to the capacity identification task, we evaluate the robustness of ICMAPE-TD3 under different priors over distractor intensity and different distances between the distractor and target location. For distractor intensity, we replace the uniform prior with a scaled Beta distribution parameterized by (α, β), using the same set of parameter values as above. When $\alpha > \beta$, the prior places more mass on higher distractor intensities, leading to stronger distractions while still achieving an average accuracy between 0.8 and 0.9; when $\alpha \leq \beta$, the accuracy remains around 0.9.

We also evaluate robustness to the Manhattan distance between the distractor and the true signal source. Across 10,000 sampled instances, ICMAPE-TD3 remains robust: accuracy improves as the distractor moves farther away and stays above 0.85. Figures 12 and 13 in the appendix summarize robustness to distance and distractor-intensity priors, respectively. Additional analyses for both tasks are provided in Appendix B.9.

6.3. Nitrate Concentration

We evaluate the proposed method on the Maryland nitrate concentration environment, where annual nitrate fields are generated using Gaussian Process (GP) models fitted to historical nitrate observations. For each year, we sample 100 GP-generated nitrate maps, with 1000 spatial points sampled from each map, and report the average performance for identifying the most concentrated regions.

Figure 6 reports the correct identification accuracy and the average number of samples. ICMAPE-TD3 achieves accuracy close to the target level on both training and test years, with only a small performance gap between them. This suggests that the learned policy is robust to GP-generated nitrate fields from unseen years. In contrast, the independent, VDN, and uniform baselines show larger drops in test-year performance, indicating weaker generalization. The GP-prior baseline achieves the highest accuracy by directly using

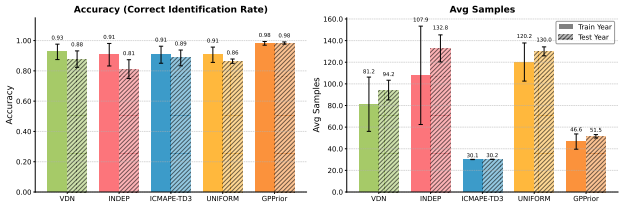


Figure 6. Evaluation on GP-generated Maryland nitrate concentration fields. Final performance comparison across methods on training years (2001–2018) and test years (2019–2023), averaged over 100 sampled GP maps per year. This plot shows estimated probability of correct identification (target is 0.9) and the average number of samples used (lower is better). Error bars indicate one standard deviation.

spatial prior information.

In terms of sample efficiency, ICMAPE-TD3 requires the fewest samples among all compared methods on both training and test years. Although the GP-prior baseline reaches slightly higher accuracy, it uses more samples than ICMAPE-TD3. Overall, ICMAPE-TD3 provides the best trade-off between accuracy and sampling cost, achieving near-target identification performance while maintaining the lowest sample complexity and stable generalization to held-out years.

7. Conclusion

In this work, we study a decentralized multi-agent pure exploration problem, where the objective is not reward maximization, but efficient information acquisition for identifying unknown properties of the environment. While existing pure exploration methods provide strong foundations, they are often tailored to specific statistical models or assume centralized action selection, leaving a gap in general learning-based methods for decentralized multi-agent data collection. We propose ICMAPE to fill this gap, a method that combines a centralized inference network with decentralized data collection policies. The inference network estimates a posterior distribution over hypotheses from global trajectory data, while each agent selects actions using only its local observation history. We evaluate ICMAPE-TD3 on two synthetic benchmarks—bandit capacity identification and signal source localization—and on a real-world Maryland nitrate concentration monitoring task, demonstrating its ability to transfer to real-world observation noise and non-stationary dynamics. We reformulate the fixed-confidence optimization problem as a learnable reinforcement learning objective and train the resulting model under the CTDE paradigm. In particular, we adapt TD3 by using a centralized critic that conditions on global trajectories and joint actions, while maintaining decentralized actors for execution. Future research could extend this setting toward more general forms of environment understanding. We discuss limitations and broader impacts in Section D.

References

Ackermann, J., Gabler, V., Osa, T., and Sugiyama, M. Reducing overestimation bias in multi-agent domains using double centralized critics, 2019. URL https://arxiv.org/abs/1910.01465.

Biewald, L. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

Chen, Z., Karthik, P. N., Tan, V. Y. F., and Chee, Y. M. Federated best arm identification with heterogeneous clients. *IEEE Transactions on Information Theory*, 70 (6):4258–4279, June 2024. ISSN 1557-9654. doi: 10.1109/tit.2023.3338027. URL http://dx.doi.org/10.1109/TIT.2023.3338027.

Degenne, R., Koolen, W. M., and Ménard, P. Non-asymptotic pure exploration by solving games. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in neural information processing systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/8d1de7457fa769ece8d93a13a59c8552-Paper.pdf.

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

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Garivier, A. and Kaufmann, E. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, pp. 998–1027. PMLR, 2016.

Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., et al. Array programming with numpy. *nature*, 585(7825):357–362, 2020.

Hu, X. and Pacchiano, A. Meet me at the arm: The cooperative multi armed bandits problem with shareable arms. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026. URL https://openreview.net/forum?id=ElXIobrn5n.

Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007.

Iqbal, S. and Sha, F. Actor-attention-critic for multi-agent reinforcement learning. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2961–2970. PMLR, 09–15 Jun 2019.

Jourdan, M., Degenne, R., Baudry, D., de Heide, R., and Kaufmann, E. Top two algorithms revisited. *Advances in Neural Information Processing Systems*, 35:26791–26803, 2022.

Kaufmann, E., Cappé, O., and Garivier, A. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17 (1):1–42, 2016.

Kota, S. R., Karthik, P., and Tan, V. Y. Almost cost-free communication in federated best arm identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8378–8385, 2023.

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

Mao, G., Fidan, B., and Anderson, B. D. Wireless sensor network localization techniques. *Computer Networks*, 51(10):2529–2553, 2007. ISSN 1389-1286. doi: https://doi.org/10.1016/j.comnet.2006.11.018. URL https://www.sciencedirect.com/science/article/pii/S1389128606003227.

Matignon, L., Laurent, G. J., and Le Fort-Piat, N. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *The Knowledge Engineering Review*, 27(1):1–31, 2012. doi: 10.1017/S0269888912000057.

McKinney, W. Data structures for statistical computing in python. *SciPy 2010*, 2010. doi: 10.25080/Majora-92bf1922-00a. URL https://doi.org/10.25080/Majora-92bf1922-00a.

Mitra, A., Hassani, H., and Pappas, G. Exploiting heterogeneity in robust federated best-arm identification. arXiv preprint arXiv:2109.05700, 2021. URL https://api.semanticscholar.org/CorpusID:237513934.

Naghshvar, M. and Javidi, T. Active Sequential Hypothesis Testing. *The Annals of Statistics*, 41(6):2703–2738, 2013. ISSN 0090-5364. URL https://www.jstor.org/stable/23566746.

National Water Quality Monitoring Council. Water quality portal, 2024. URL https://www.waterqualitydata.us/. Accessed: June 26, 2026.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

Patwari, N., Ash, J., Kyperountas, S., Hero, A., Moses, R., and Correal, N. Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine*, 22(4):54–69, 2005. doi: 10.1109/MSP.2005.1458287.

Paul, A. K. and Sato, T. Localization in wireless sensor networks: A survey on algorithms, measurement techniques, applications and challenges. *Journal of Sensor and Actuator Networks*, 6(4), 2017. ISSN 2224-2708. doi: 10.3390/jsan6040024. URL https://www.mdpi.com/2224-2708/6/4/24.

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

Russo, A. and Pacchiano, A. Adaptive Exploration for Multi-Reward Multi-Policy Evaluation. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 52382–52421. PMLR, October 2025. URL https://proceedings.mlr.press/v267/russo25a.html.

Russo, A., Song, Y., and Pacchiano, A. Pure exploration with feedback graphs. In *Proceedings of the 28th international conference on artificial intelligence and statistics*, Proceedings of machine learning research. PMLR, 2025.

Russo, A., Lee, Y.-C., Welch, R., and Pacchiano, A. In-context learning for pure exploration in continuous spaces, 2026a. URL https://arxiv.org/abs/2602.17976.

Russo, A., Welch, R., and Pacchiano, A. In-context learning for pure exploration. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=NFCNFvDYfE.

Russo, D. Simple Bayesian Algorithms for Best Arm Identification. In *Conference on Learning Theory*, pp. 1417–1418. PMLR, June 2016. URL https://proceedings.mlr.press/v49/russo16.html.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Sheikh, H. U. and Bölöni, L. Multi-agent reinforcement learning for problems with combined individual and team reward. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2020.

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

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

Szostak, H. and Cohen, K. Deep multi-agent reinforcement learning for decentralized active hypothesis testing. *IEEE Access*, 12:130444–130459, 2024.

Tan, M. Multi-agent reinforcement learning: independent versus cooperative agents. In *Proceedings of the Tenth International Conference on International Conference on Machine Learning*, ICML'93, pp. 330–337, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc. ISBN 1558603077.

Towers, M., Kwiatkowski, A., Terry, J., Balis, J. U., De Cola, G., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

U.S. Environmental Protection Agency. Estimated nitrate concentrations in groundwater used for drinking, 2026. URL https://www.epa.gov/nutrientpollution/estimated-nitrate-concentrations-groundwater-used-drinking.

U.S. Geological Survey. Nutrients and eutrophication, 2026. URL https://www.usgs.gov/mission-areas/water-resources/science/nutrients-and-eutrophication.

Vannella, F., Proutiere, A., and Jeong, J. Best arm identification in multi-agent multi-armed bandits. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Vannella, F., Proutiere, A., Jedra, Y., and Jeong, J. Learning optimal antenna tilt control policies: A contextual linear bandits approach. *IEEE Transactions on Mobile Computing*, 23(12):12666–12679, 2024. doi: 10.1109/TMC.2024.3424192.

Wang, P.-A., Tzeng, R.-C., and Proutiere, A. Fast Pure Exploration via Frank-Wolfe. In *Advances in Neural Information Processing Systems*, volume 34, pp. 5810–5821. Curran Associates, Inc., 2021.

Wang, X., Xie, H., and Lui, J. C. S. Multiple-Play Stochastic Bandits with Shareable Finite-Capacity Arms. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 23181–23212. PMLR, 2022a.

Wang, X., Xie, H., and Lui, J. C. S. Multi-player multi-armed bandits with finite shareable resources arms: Learning algorithms and applications. In Raedt, L. D. (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 3537–3543. International Joint Conferences on Artificial Intelligence Organization, 7 2022b. doi: 10.24963/ijcai.2022/491. URL https://doi.org/10.24963/ijcai.2022/491.

Waskom, M. L. Seaborn: statistical data visualization. *Journal of open source software*, 6(60):3021, 2021.

Wei, Y., Shan, X., Miao, R., and Li, J. Lero: Llm-driven evolutionary framework with hybrid rewards and enhanced observation for multi-agent reinforcement learning. In *International Conference on Intelligent Computing*, pp. 15–26. Springer, 2025.

Yadan, O. Hydra - a framework for elegantly configuring complex applications. Github, 2019. URL https://github.com/facebookresearch/hydra.

Zhang, Y. and Guo, D. Multi-agent reinforcement learning for multi-cell spectrum and power allocation. *IEEE Transactions on Communications*, 73(8):5980–5992, 2025.

A. Theoretical Details

Our theoretical formulation follows from the In-Context Pure Exploration (Russo et al., 2026b).

A.1. Preliminaries

Assumption A.1 [Common dominating measures]. *There exist reference measures λ_0 and λ on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ such that, for all $(\rho, P) \in \mathcal{M}, s \in \mathbb{N}$, and $(z, a) \in \mathcal{Z} \times \mathcal{A}$,*

$$\rho(\cdot) \ll \lambda_0(\cdot) \quad \textit{and} \quad P_s(\cdot \mid z, a) \ll \lambda(\cdot).$$

That is, all initial distributions and transition kernels have densities with respect to common reference measures:

$$\rho^\omega(dx) = p_0^\omega(x)\lambda_0(dx), \qquad P_s^\omega(dx \mid z, a) = p_s^\omega(x \mid z, a)\lambda(dx).$$

Lemma A.1 (Posterior kernel under decentralized policies)**.** *Fix a time $t \in \mathbb{N}$. Consider a model class $\mathcal{M}$ with prior $\mathcal{P}$, and let $M \sim \mathcal{P}$ denote the unknown environment. Under a decentralized joint policy*

$$\pi = (\pi^1, \ldots, \pi^N), \qquad \pi_s^i(\cdot \mid \mathcal{D}_s^i) \in \Delta(\mathcal{A}^i),$$

each agent i selects action a_s^i using only its local trajectory $\mathcal{D}_s^i$. Let the global trajectory up to time t be

$$\mathcal{D}_t = (x_1, a_1, x_2, a_2, \ldots, x_t),$$

where $x_s = (x_s^1, \ldots, x_s^N)$ and $a_s = (a_s^1, \ldots, a_s^N)$.

Then there exists a probability kernel

$$R_t : \mathcal{D}_t \times \mathcal{B}(\mathcal{M}) \to [0, 1],$$

independent of the decentralized policy π, such that for every measurable set $A \subseteq \mathcal{M}$ and every measurable trajectory event Z,

$$\mathbb{P}^\pi(M \in A, \mathcal{D}_t \in Z) = \int_Z R_t(A \mid z)\, \mathbb{P}^\pi(\mathcal{D}_t \in dz).$$

Consequently, for any hypothesis set $B \subseteq \mathcal{H}$,

$$\mathbb{P}^\pi(H^\star \in B \mid \mathcal{D}_t = z) = R_t\big(\{M \in \mathcal{M} : h^\star(M) \in B\} \mid z\big),$$

for $\mathbb{P}^\pi$-almost every trajectory z.

Proof. Let $z = (x_1, a_1, x_2, \ldots, a_{t-1}, x_t) \in \mathcal{Z}_t$ denote a realized global trajectory. Under a decentralized policy π, the trajectory law induced by model M factorizes as

$$\mathbb{P}_{M,t}^\pi(dz) = \rho_M(dx_1) \prod_{s=1}^{t-1} \left[\prod_{i=1}^{N} \pi_s^i(da_s^i \mid \mathcal{D}_s^i) \right] P_{M,s}(\mathrm{d}x_{s+1} \mid \mathcal{D}_s, a_s).$$

The policy-dependent factor

$$\prod_{s=1}^{t-1} \prod_{i=1}^{N} \pi_s^i(a_s^i \mid \mathcal{D}_s^i)$$

depends on the realized trajectory z, but not on the model M. Hence, once we condition on the trajectory, the action-selection probabilities do not provide additional likelihood information about M

We choose reference measures λ_0 and λ such that

$$\rho_M(dx_1) = r_M(x_1)\lambda_0(dx_1), \qquad P_{M,s}(dx_{s+1} \mid \mathcal{D}_s, a_s) = p_{M,s}(x_{s+1} \mid \mathcal{D}_s, a_s)\lambda(dx_{s+1}).$$

Define a reference measure on the trajectory space by

$$\nu_t^\pi(dz) := \lambda_0(dx_1) \prod_{s=1}^{t-1} \left[\prod_{i=1}^{N} \pi_s^i(da_s^i \mid \mathcal{D}_s^i) \right] \lambda(dx_{s+1}).$$

Then $\mathbb{P}_{M,t}^\pi \ll \nu_t^\pi$ by Assumption A.1, and the Radon–Nikodym derivative is

$$\ell_t(M, z) := \frac{d\mathbb{P}_{M,t}^\pi}{d\nu_t^\pi}(z) = r_M(x_1) \prod_{s=1}^{t-1} p_{M,s}(x_{s+1} \mid \mathcal{D}_s, a_s).$$

The policy terms are absorbed into the reference measure ν_t^π, so the likelihood $\ell_t(M, z)$ depends only on the model-dependent initial density, transition densities, and observed trajectory, but not on the decentralized policy.

Therefore, for any measurable $A \subseteq \mathcal{M}$ and $Z \subseteq \mathcal{Z}_t$,

$$\mathbb{P}^\pi(M \in A, \mathcal{D}_t \in Z) = \int_A \int_Z \ell_t(M, z)\, \nu_t^\pi(dz)\, \mathcal{P}(dM),$$

and

$$\mathbb{P}^\pi(\mathcal{D}_t \in Z) = \int_Z \int_{\mathcal{M}} \ell_t(M, z)\, \mathcal{P}(dM)\, \nu_t^\pi(dz).$$

Thus the posterior kernel is

$$R_t(A \mid z) := \frac{\int_A \ell_t(M, z)\, \mathcal{P}(dM)}{\int_{\mathcal{M}} \ell_t(M, z)\, \mathcal{P}(dM)}.$$

This kernel is independent of π. Equivalently,

$$\mathbb{P}^\pi(M \in A \mid \mathcal{D}_t = z) = R_t(A \mid z),$$

for $\mathbb{P}^\pi$-almost every z. The desired conclusion follows by taking

$$A = \{M \in \mathcal{M} : h^\star(M) \in B\}.$$

□

Proposition A.1. *Consider a fixed decentralized policy $\pi \in \prod_{dec}$. Let $t \in \mathbb{N}$, and $z \sim \mathbb{P}_t^\pi$. For any t, the optimal inference rule of $\sup_{\mathcal{I}_t} \mathbb{P}_t^\pi(H^\star = \mathcal{I}_t(\mathcal{D}_t))$ is given by $\mathcal{I}_t^*(z) = \arg\max_{H \in \mathcal{H}} \mathbb{P}_t^\pi(H^\star = H \mid \mathcal{D}_t = z)$*

The proof of Proposition A.1 follows from Proposition B.2 of Russo et al. (2026b).

A.2. Optimization Problem

Fixed Confidence: Given a target error $\delta \in (0, 1)$, the learner chooses: a stopping time $\tau \in \mathbb{N}$, which is the total number of queries; a data collection policy π; and an inference rule $\mathcal{I}$ that minimize the expected total number of queries τ (see Assumption 4.1), while meeting the correctness guarantee. τ marks the random moment when the data collection process stops. We define the following optimization problem:

$$\min_{\{\pi^i\}, \tau, \mathcal{I}} \mathbb{E}^\pi[\tau] \quad \textbf{s.t.} \quad \mathbb{P}^{\{\pi^i\}}(\mathcal{I}_\tau(\mathcal{D}_\tau) = H^\star) \geq 1 - \delta,$$

where $\mathbb{P}^\pi(\cdot)$ is the probability of the underlying data collection process when π gathers data from environment M, and M is sampled from the prior $\mathcal{P}$. In the appendix, we use the notation $\{\pi^i\}$ to represent policies in the decentralized policy class Π_{dec}.

Decentralized policy class. Since agents operate under decentralized observation, we restrict to decentralized policies Π_{dec} such that for every agent i and time t, $a_i(t)$ depends only on its own local observation history $\mathcal{D}_{t-1}^i$ and is conditionally independent of all other agents' histories given its own:

$$a_i(t) \sim \pi_i(\cdot \mid \mathcal{D}_{t-1}^i).$$

Furthermore, π_i is measurable with respect to $\sigma(\mathcal{D}_{t-1}^i)$.

A.3. Dual Problem Formulation and RL Conversion

Dual Problem Formulation In the fixed confidence setting, we are solving the following optimization problem.

$$\begin{aligned} \inf_{\pi \in \Pi_{\mathrm{dec}},\, \tau,\, I} \quad & \mathbb{E}^{\pi}[\tau] \\ \text{s.t.} \quad & \mathbb{P}^{\pi}(I_\tau(\mathcal{D}_\tau) = H^\star) \geq 1-\delta, \\ & \mathbb{E}^{\pi}[\tau] < \infty, \end{aligned} \tag{7}$$

where we assumed τ to be locally adapted to $(\sigma(\mathcal{D}_t^i))_t$ according to Assumption 4.1. Here $\mathcal{D}_t^i = \{(i, a_s^i, x_s^i)\}_{s=1}^t$, denotes the local state of player i, i.e., the sequence of tuples containing the player index together with its local observation–action trajectory up to time t.

Lemma A.2. *The Lagrangian dual of the problem in Eq. 7 is given by*

$$\sup_{\lambda \geq 0} \inf_{\{\pi^i\},\tau,\mathcal{I}} \mathcal{L}(\{\pi^i\}, \tau, \mathcal{I}; \lambda) = \sup_{\lambda \geq 0} \inf_{\pi,\tau,\mathcal{I}} \lambda(1-\delta) + \mathbb{E}^{\{\pi^i\}}\Big[\tau - \lambda \mathbb{P}_{\{\pi^i\}}(H^\star = \mathcal{I}_\tau(\mathcal{D}_\tau) \mid \mathcal{D}_\tau\Big],$$

where λ is the Lagrange multiplier.

Proof. Since any feasible solution stops almost surely under assumption, we use $\{\pi^i\}$ to mean decentralized policies and get

$$\begin{aligned} \mathbb{P}^{\{\pi^i\}}(H^\star = \mathcal{I}_\tau(\mathcal{D}_\tau)) &= \sum_{t \geq 1} \mathbb{P}_t^{\{\pi^i\}}(\mathcal{I}_t(\mathcal{D}_t) = H^\star,\ \tau = t) \qquad \text{(law of total probability)} && (8) \\ &= \sum_{t \geq 1} \mathbb{E}_t^{\{\pi^i\}}\left[\mathbf{1}_{\{\tau=t\}} \mathbf{1}_{\{H^\star = \mathcal{I}_t(\mathcal{D}_t)\}}\right] && (9) \\ &= \sum_{t \geq 1} \mathbb{E}_t^{\{\pi^i\}}\left[\mathbb{E}_t[\mathbf{1}_{\{\tau=t\}} \mathbf{1}_{\{H^\star = \mathcal{I}_t(\mathcal{D}_t)\}} \mid \mathcal{D}_t]\right], \qquad \text{(tower rule)} && (10) \\ &= \sum_{t \geq 1} \mathbb{E}_t^{\{\pi^i\}}\left[\mathbf{1}_{\{\tau=t\}} \mathbb{E}_t^{\{\pi^i\}}\left[\mathbf{1}_{\{H^\star = \mathcal{I}_t(\mathcal{D}_t)\}} \mid \mathcal{D}_t\right]\right] \qquad (\{\tau = t\} \in \sigma(\mathcal{D}_t^i)) && (11) \\ &= \sum_{t \geq 1} \mathbb{E}_t^{\{\pi^i\}}\left[\mathbf{1}_{\{\tau=t\}} \mathbb{P}_t(H^\star = \mathcal{I}_t(\mathcal{D}_t) \mid \mathcal{D}_t)\right]. && (12) \end{aligned}$$

The last line holds because of Lemma A.1. Then we introduce a Lagrange multiplier $\lambda \geq 0$. The corresponding Lagrangian function is given by

$$\mathcal{L}(\{\pi^i\}, \tau, I; \lambda) = \mathbb{E}^{\{\pi^i\}}[\tau] + \lambda\Big((1-\delta) - \mathbb{P}^{\{\pi^i\}}(I_\tau(\mathcal{D}_\tau) = H^\star)\Big).$$

The dual function $g(\lambda)$ is defined as

$$g(\lambda) = \inf_{\{\pi^i\},\tau,\mathcal{I}} \mathcal{L}(\{\pi^i\}, \tau, \mathcal{I}; \lambda).$$

The dual problem is

$$\sup_{\lambda \geq 0} g(\lambda) = \sup_{\lambda \geq 0} \quad \inf_{\{\pi^i\},\tau,\mathcal{I}} \mathcal{L}(\{\pi^i\}, \tau, \mathcal{I}; \lambda) \tag{13}$$

To simplify the Lagrangian function we get:

$$\mathcal{L}(\{\pi^i\}, \tau, \mathcal{I}; \lambda) = \mathbb{E}^{\{\pi^i\}}[\tau] + \lambda\Big((1-\delta) - \mathbb{P}^{\{\pi^i\}}(I_\tau(\mathcal{D}_\tau) = H^\star)\Big) \tag{14}$$

$$= \mathbb{E}^{\{\pi^i\}}[\tau] + \lambda\Big((1-\delta) - \sum_{t\geq 1}\mathbb{E}^{\{\pi^i\}}\big[\mathbf{1}_{\{\tau=t\}}\mathbb{P}_t(H^\star = \mathcal{I}_t(\mathcal{D}_t) \mid \mathcal{D}_t)\big]\Big) \tag{15}$$

$$= \lambda - \lambda\delta + \mathbb{E}^{\{\pi^i\}}\left[\tau - \lambda\sum_{t\geq 1}\mathbf{1}_{\{\tau=t\}}\mathbb{P}_t(H^\star = \mathcal{I}_t(\mathcal{D}_t) \mid \mathcal{D}_t)\Big)\right] \tag{16}$$

$$= \lambda - \lambda\delta + \mathbb{E}^{\{\pi^i\}}\left[\sum_{t\geq 1}\mathbf{1}_{\{t\leq\tau\}} - \lambda\sum_{t\geq 1}\mathbf{1}_{\{\tau=t\}}\mathbb{P}_t(H^\star = \mathcal{I}_t(\mathcal{D}_t) \mid \mathcal{D}_t)\Big)\right] \tag{17}$$

$$= \lambda - \lambda\delta + \mathbb{E}^{\{\pi^i\}}\left[\tau - \lambda\mathbb{P}_\tau(H^\star = \mathcal{I}_\tau(\mathcal{D}_\tau) \mid \mathcal{D}_\tau)\Big)\right] \tag{18}$$

□

Now we show how to embed the stopping rule into the policy by augmenting the action space as a stopping action. At each time t, the joint action is $a_t = (a_t^1, \ldots, a_t^N) \in \mathcal{A}$, where each agent chooses $a_t^i \in \mathcal{A}^i$.

For each agent $i \in [N]$, define the augmented local action space

$$\bar{\mathcal{A}}^i := \mathcal{A}^i \cup \{a_{\text{stop}}\},$$

and the augmented joint action space

$$\bar{\mathcal{A}} := \bar{\mathcal{A}}^1 \times \cdots \times \bar{\mathcal{A}}^N.$$

We also define the induced stopping time under the augmented policy as

$$\bar{\tau} := \inf\big\{t \geq 1 : \ \exists i \in [N] \text{ s.t. } a_t^i = a_{\text{stop}}\big\}.$$

If any agent performed action $a_t^i \neq a_{\text{stop}}$ and observed x_{t+1}^i after its action decides to stop this is equivalent to choose action a_{stop} at round $t+1$ leading to $\bar{\tau} = t+1 = \tau$.

Lemma A.3. *For every $(\{\pi^i\}, \tau, \mathcal{I})$ with $\tau \geq 1$ almost surely, there exists a policy $\{\bar{\pi}^i\}$ on $\bar{\mathcal{A}} = \bigcup_i(\mathcal{A}^i \cup \{a_{stop}\})$ with stopping rule*

$$\bar{\tau} := \inf\big\{t \geq 1 : \ \exists i \in [N] \textit{ s.t. } a_t^i = a_{stop}\big\}.$$

such that

$$\mathcal{L}(\{\pi^i\}, \tau, \mathcal{I}; \lambda) = \mathcal{L}(\{\bar{\pi}^i\}, \mathcal{I}; \lambda).$$

Proof. With Assumption 4.1 $\{\tau \leq t\} \in \sigma(\mathcal{D}_t^i) \quad \forall i$, we can define for each agent for $z^i \in \mathcal{Z}_t^i$,

and there exists a measurable set $A_t^i \in \mathcal{B}(\mathcal{Z}_t^i)$, where $\{\tau = t\} = \{\mathcal{D}_t^i \in A_t^i\}$

$$\bar{\pi}_t^i(a_{\text{stop}} \mid z^i) = \mathbf{1}\{z^i \in A_t^i\} \qquad \text{and} \qquad \bar{\pi}_t^i(a \mid z^i) = \pi_t^i(a \mid z^i), \quad a \in \mathcal{A}^i.$$

By the construction of the $\{\bar{\pi}^i\}$,

$$a_{\text{stop}} \Longleftrightarrow \tau = t \text{ for all } i \in [N].$$

Hence,

$$\bigcup_i \{a_t^i = a_{\text{stop}}\} = \{\tau = t\}$$

Therefore, we have

$$\{\bar{\tau} = t\} = \{\exists i : a_t^i = a_{\text{stop}}\} = \{\tau = t\}.$$

$$\mathcal{L}(\{\bar{\pi}^i\}, \mathcal{I}; \lambda) = \lambda(1-\delta) + \mathbb{E}^{\{\bar{\pi}^i\}}\left[\bar{\tau} - \lambda \mathbb{P}_{\bar{\tau}}(H^\star = \mathcal{I}_{\bar{\tau}}(\mathcal{D}_{\bar{\tau}}) \mid \mathcal{D}_{\bar{\tau}})\right] \tag{19}$$

$$= \lambda(1-\delta) + \mathbb{E}^{\{\bar{\pi}^i\}}\left[\sum_{t\geq 1} \mathbf{1}_{\{t\leq\bar{\tau}\}} - \lambda \mathbf{1}_{\{\bar{\tau}=t\}} \mathbb{P}_t(H^\star = \mathcal{I}_t(\mathcal{D}_t) \mid \mathcal{D}_t)\right] \tag{20}$$

$$= \lambda(1-\delta) + \mathbb{E}^{\{\bar{\pi}^i\}}\left[\sum_{t\geq 1}^{\bar{\tau}} 1 - \lambda \mathbf{1}_{\{\bar{\tau}=t\}} \mathbb{P}_t(H^\star = \mathcal{I}_t(\mathcal{D}_t) \mid \mathcal{D}_t)\right] \tag{21}$$

$$= \lambda(1-\delta) + \mathbb{E}^{\{\bar{\pi}^i\}}\left[\sum_{t\geq 1}^{\bar{\tau}} 1 - \lambda \mathbf{1}_{\{\exists i: a_t^i = a_{\text{stop}}\}} \mathbb{P}_t(H^\star = \mathcal{I}_t(\mathcal{D}_t) \mid \mathcal{D}_t)\right] \tag{22}$$

$$= \lambda(1-\delta) + \mathbb{E}^{\{\pi^i\}}\left[\sum_{t\geq 1}^{\tau} 1 - \lambda \mathbf{1}_{\{\tau=t\}} \mathbb{P}_t(H^\star = \mathcal{I}_t(\mathcal{D}_t) \mid \mathcal{D}_t)\right] \tag{23}$$

□

Now we will keep working in the extended action space, and define $\tau := \inf\left\{t \geq 1 : \exists i \in [N] \text{ s.t. } a_t^i = a_{\text{stop}}\right\}$ to simply notations by avoiding using the bar notation.

Proposition A.2. *For any $t \in \mathbb{N}$, the optimal inference rule selects the hypothesis with the largest posterior probability:*

$$\mathcal{I}_t(D_t) \in \arg\max_{H\in\mathcal{H}} \mathbb{P}_t(H^\star = H \mid \mathcal{D}_t),$$

Moreover,

$$\sup_{\lambda\geq 0} \inf_{\pi\in\Pi_{dec},\mathcal{I}} \mathcal{L}_\lambda(\pi, \mathcal{I}) = \sup_{\lambda\geq 0} \inf_{\pi} \left\{\lambda(1-\delta) + \mathbb{E}^{\pi}[\tau - \lambda r_\tau(\mathcal{D}_\tau)]\right\},$$

where

$$\tau := \inf\left\{t \geq 1 : \exists i \in [N] \textit{ s.t. } a_t^i = a_{\textit{stop}}\right\} \qquad r_t(z_t) := \max_{H\in\mathcal{H}} \mathbb{P}_t(H^\star = H \mid \mathcal{D}_t = z_t).$$

Proof. Let's first optimizing over the inference rules. For $t \in \mathbb{N}, z \in \mathcal{Z}_t$. let's define $r_t(z_t) = \max_{H\in\mathcal{H}} \mathbb{P}_t(H^* = H \mid \mathcal{D}_t = z_t)$. If we fix $(\{\pi^i\}, \tau)$, we will get

$$\inf_{\mathcal{I}} \mathcal{L}(\{\pi^i\}, \mathcal{I}; \lambda) = \lambda(1-\delta) + \inf_{\mathcal{I}} \mathbb{E}^{\{\pi^i\}}\left[\sum_{t\geq 1} \mathbf{1}_{\{t\leq\tau\}} - \lambda \mathbf{1}_{\{\tau=t\}} \mathbb{P}_t(H^\star = \mathcal{I}_t(\mathcal{D}_t) \mid \mathcal{D}_t)\right],$$

$$= \lambda(1-\delta) + \mathbb{E}^{\{\pi^i\}}\left[\inf_{\mathcal{I}} \sum_{t\geq 1} \mathbf{1}_{\{t\leq\tau\}} - \lambda \mathbf{1}_{\{\tau=t\}} \mathbb{P}_t(H^\star = \mathcal{I}_t(\mathcal{D}_t) \mid \mathcal{D}_t)\right],$$

$$= \lambda(1-\delta) + \mathbb{E}^{\{\pi^i\}}\left[\sum_{t\geq 1} \mathbf{1}_{\{t\leq\tau\}} - \lambda \mathbf{1}_{\{\tau=t\}} \sup_{\mathcal{I}_t} \mathbb{P}_t(H^\star = \mathcal{I}_t(\mathcal{D}_t) \mid \mathcal{D}_t)\right],$$

$$= \lambda(1-\delta) + \mathbb{E}^{\{\pi^i\}}\left[\sum_{t\geq 1} \mathbf{1}_{\{t\leq\tau\}} - \lambda \mathbf{1}_{\{\tau=t\}} r_t(\mathcal{D}_t)\right]$$

$$= \lambda(1-\delta) + \mathbb{E}^{\{\pi^i\}}\left[\tau - \lambda r_t(\mathcal{D}_t)\right].$$

The second-to-last equality follows from Proposition A.1. Now we can get

$$\sup_{\lambda\geq 0} \inf_{\{\pi^i\},\mathcal{I}} \mathcal{L}(\{\pi^i\}, \mathcal{I}; \lambda) = \sup_{\lambda\geq 0} \inf_{\{\pi^i\}} \lambda - \lambda\delta + \mathbb{E}^{\{\pi^i\}}\left[\tau - \lambda r_\tau(\mathcal{D}_\tau)\right]$$

□

We now optimize over policies. Recall we now defined $\mathcal{A}^i$ includes the stopping action, $a \in \mathcal{A}$ and $\tau = \inf\{t : a_t = a_{\text{stop}}\} \Leftrightarrow \tau = \inf\left\{t \geq 1 : \exists i \in [N] \text{ s.t. } a_t^i = a_{\text{stop}}\right\}$. For $t \in \mathbb{N}$, $z \in \mathcal{Z}_t$ use π as an simplified notation for any possible decentralized policy, then the optimal value to go satisfies

$$V_t(z;\lambda) := \inf_{\pi:\tau\geq t} \lambda - \lambda\delta + \mathbb{E}^{\pi}\left[\underbrace{(\tau - t)}_{\text{expected future sample}} - \lambda r_\tau(\mathcal{D}_\tau) \,\middle|\, \mathcal{D}_t = z\right].$$

Also define the following optimal Q-function for $z \in \mathcal{Z}_t$, $a \neq a_{\text{stop}}$

$$Q_t(z,a;\lambda) := 1 + \int_{\mathcal{X}} V_{t+1}(\underbrace{z,a,x'}_{z'};\lambda)\, \bar{P}_t(dx' \mid z,a),$$

where $\bar{P}_t$ is the posterior predictive distribution of the next observation. Every additional step will give a cost of 1 in the Q value. We also define stopping action Q value as

$$Q_t(z, a_{\text{stop}};\lambda) := \lambda\big(1 - \delta - r_t(z)\big).$$

If we define λ-optimal policy $\pi_\lambda^\star = (\pi_t^\star(z;\lambda))_t$ as

$$\pi_t^\star(z;\lambda) \in \arg\min_{a\in\mathcal{A}} Q_t(z,a;\lambda)$$

We have then the following result.

Proposition A.3. *$\pi_\lambda^\star$ is a λ-optimal policy. Furthermore, the optimal value for $z \in \mathcal{Z}_t$ satisfies*

$$V_t(z;\lambda) = \min_{a\in\mathcal{A}} Q_t(z,a;\lambda). \tag{21}$$

Proof. Fix $D_t = z$, $z \in \mathcal{Z}_t$. Assume the optimal stopping action stops at $\tau = t$ for such z. Then

$$V_t(z;\lambda) = \lambda - \lambda\delta - \lambda r_t(z) = Q_t(z, a_{\text{stop}};\lambda).$$

Otherwise, suppose the optimal stopping rule satisfies $\tau > t$, then

$$\begin{aligned}
V_t(z;\lambda) &= \inf_{\pi:\tau>t} \lambda - \lambda\delta + \mathbb{E}^{\pi}[\tau - t - \lambda r_\tau(D_\tau) \mid D_t = z] \\
&= \inf_{\pi:\tau>t} \lambda - \lambda\delta + 1 + \mathbb{E}^{\pi}[\tau - (t+1) - \lambda r_\tau(D_\tau) \mid D_t = z] \\
&= \inf_{\pi:\tau>t} 1 + \mathbb{E}^{\pi}\left[\mathbb{E}^{\pi}\left[\lambda - \lambda\delta + \tau - (t+1) - \lambda r_\tau(D_\tau) \mid D_{t+1}\right] \mid \mathcal{D}_t = z\right] \\
&= \inf_{\pi:\tau>t} 1 + \mathbb{E}^{\pi}\left[\mathbb{E}^{\pi}\left[V_{t+1}(\mathcal{D}_{t+1};\lambda)\right] \mid \mathcal{D}_t = z\right] \\
&= \inf_{\pi:\tau>t} 1 + \mathbb{E}_a^{\pi}\left[\int_{\mathcal{X}} V_{t+1}(z,a,x';\lambda)\, \bar{P}_t(dx'|z,a)\right],
\end{aligned}$$

We then clearly obtain the lower bound

$$V_t(z;\lambda) \geq \min\left\{\lambda(1 - \delta - r_t(z)),\ \min_{a\in A} Q_t(z,a;\lambda)\right\}.$$

If $\tau = t$, then $\lambda(1 - \delta - r_t(z)) \leq \min_{a\neq\text{stop}} Q_t(z,a;\lambda)$; the value to go for the optimal policy is exactly $\lambda(1 - \delta - r_t(z))$.

If $\tau \neq t$, then $\lambda(1 - \delta - r_t(z)) \geq \min_{a\neq\text{stop}} Q_t(z,a;\lambda)$, the best policy value should go to $Q_t(z,\pi_t(z);\lambda) = \min_{a\in\mathcal{A}} Q_t(z,a,\lambda) = V(z;\lambda)$. □

Reward-form dual. Equivalently, the constrained problem can be written in reward-maximization form as

$$\inf_{\lambda \geq 0} \sup_{\pi, \mathcal{I}} V_\lambda(\pi, \mathcal{I}),$$

where

$$V_\lambda(\pi, \mathcal{I}) := -\mathbb{E}^\pi[\tau] + \lambda\Big(\mathbb{P}^\pi(\mathcal{I}_\tau(\mathcal{D}_\tau) = H^\star) - 1 + \delta\Big).$$

Optimizing over the inference rule gives

$$r_t(\mathcal{D}_t) := \max_{H \in \mathcal{H}} \mathbb{P}_t(H^\star = H \mid \mathcal{D}_t),$$

and therefore

$$\inf_{\lambda \geq 0} \sup_{\pi} \mathbb{E}^\pi\left[-\tau + \lambda\big(r_\tau(\mathcal{D}_\tau) - 1 + \delta\big)\right].$$

We then define the reward model by

$$r_{t,\lambda}(\mathcal{D}_t, a) = -\mathbf{1}\{a \neq a_{\text{stop}}\} + \lambda\, \mathbf{1}\{a = a_{\text{stop}}\}\big(\max_{H \in \mathcal{H}} \mathbb{P}_t(H^\star = H \mid \mathcal{D}_t)\big).$$

Accordingly, the Q-function satisfies

$$Q_{t,\lambda}(\mathcal{D}_t, a) = r_{t,\lambda}(\mathcal{D}_t, a) + \mathbf{1}\{a \neq a_{\text{stop}}\}\mathbb{E}_{x_{t+1}|(\mathcal{D}_t, a)}\left[\max_{a'} Q_{t+1,\lambda}\big((\mathcal{D}_t, a, x_{t+1}), a'\big)\right].$$

A.4. Q-Learning Approach

From the dynamic programming formulation above, we obtain the optimal joint action–value function associated with the optimal policy $\pi^\star$. We denote this value by Q_{tot}. We next seek a decomposition of Q_{tot} into per-agent utilities Q^i. To enable decentralized greedy action selection, we adopt the *individual–global maximization* (IGM) assumption, which ensures that the optimal joint action can be recovered from individual greedy actions. In particular, we assume there exists a mixing function such that Q_{tot} is monotone in each Q^i, i.e.,

$$\frac{\partial Q_{\text{tot}}}{\partial Q^i} \geq 0, \qquad \forall i.$$

Under this condition, decentralized greedy selection with respect to the individual utilities is consistent with the joint optimum. In practice, we use learned approximations $\hat{Q}^i$ to parameterize this decomposition, rather than the true value functions.

Since τ is a common stopping time adapted to each local filtration in Assumption 4.1, on the event $\{\tau = t\}$ all agents must stop, i.e., $a_t^i = a_{\text{stop}}$ for all i. Because decentralized execution is greedy with respect to the learned per-agent utilities, we impose the following consistency constraint: on $\{\tau = t\}$, we require

$$\Big(\hat{Q}_t^i(z_t^i, a_{\text{stop}}; \lambda) \geq \hat{Q}_t^i(z_t^i, a; \lambda)\Big), \qquad \forall a \in \mathcal{A}^i,\ a \neq a_{\text{stop}}.$$

Equivalently,

$$a_{\text{stop}} \in \arg\max_{a^i \in \mathcal{A}^i} \hat{Q}_t^i(z_t^i, a^i; \lambda).$$

Suppose We adopt a VDN-style factorization,

$$\hat{Q}_{\text{tot},t}(z, \mathbf{a}; \lambda) = \sum_{i=1}^{N} \hat{Q}_t^i(z_t^i, a_t^i; \lambda),$$

so that joint greedy minimization is consistent with decentralized greedy minimization.

A.5. Policy Learning Approach

An alternative to Q-learning is to learn a decentralized policy using an actor–critic formulation, where the critic evaluates the actors' actions using global information and joint actions. Since execution remains decentralized, each agent i selects its action based only on its local history $\mathcal{D}_t^i$. During training, however, the critic may use additional global information to improve credit assignment.

For each agent $i \in [N]$, let the decentralized actor be

$$\pi_\theta^i(\cdot \mid \mathcal{D}_t^i),$$

and θ is the collection of actor's parameter.

Critic. For each agent i, we define a centralized training critic

$$Q_t(\mathcal{D}_t, a_t^i, a_t^{-i}; \lambda),$$

which estimates the future soft cost when agent i takes action a_t^i and the other agents take a_t^{-i}. If no agent stops at time t, we pay cost 1 and continue. If some agent stops, we terminate with reward

$$\lambda\big(r_t(\mathcal{D}_t) - 1 + \delta\big).$$

Thus, the one-step target is

$$y_t^i = c_t + \gamma\, \mathbf{1}\{\forall j,\ a_t^j \neq a_{\text{stop}}\}\Big(Q_{\phi^{\text{tar}}}(\mathcal{D}_{t+1}, a_{t+1}^i, a_{t+1}^{-i}; \lambda))\Big),$$

where

$$c_t = \begin{cases} -1, & \text{if no agent stops at time } t, \\ \lambda\big(r_t(\mathcal{D}_t) - 1 + \delta\big), & \text{otherwise.} \end{cases}$$

The critic loss is

$$\mathcal{L}_{Q^i}(\phi^i) = \mathbb{E}\Big[\big(Q_{\phi^i}^i(\mathcal{D}_t, a_t^i, a_t^{-i}; \lambda) - y_t^i\big)^2\Big].$$

Actor. The actor is trained with a deterministic policy-gradient objective using the centralized critics. Given the local history $\mathcal{D}_t^i$, each actor produces a relaxed action $\tilde{a}_t^i = \pi_{\theta_i}^i(\mathcal{D}_t^i)$, which is used for the actor update. The objective is

$$J(\theta) = \mathbb{E}\left[\sum_{i=1}^{N} \min_{j\in\{1,2\}} Q_{\phi_j}\big(\mathcal{D}_t, \tilde{a}_t^i, \tilde{a}_t^{-i}; \lambda\big)\right],$$

where $\tilde{a}_t = (\tilde{a}_t^1, \ldots, \tilde{a}_t^N)$ denotes the joint relaxed action generated by the current actors. Since we maximize $J(\theta)$, the actor is encouraged to choose decentralized actions that receive high value under the centralized critics. During execution, each agent still selects actions using only its local history $\mathcal{D}_t^i$.

Stopping. Since τ is a common stopping time, when stopping is optimal all agents should prefer the stopping action. Hence, when $\{t = \tau\}$ we want

$$a_{\text{stop}} \in \arg\max_{a^i \in \mathcal{A}^i} Q_t(\mathcal{D}_t, a^i, a_t^{-i}; \lambda), \qquad \forall i \in [N].$$

In the soft setting, this means that the policy should assign high probability to a_{stop}.

A.6. Posterior Confidence Reward for Capacity Identification

Given a bandit problem with K arms and N players, the objective is identifying the true capacity vector C^* up to $1-\delta$ confidence. We define the set of hypothesis $\mathcal{H}$ is all the possible capacity configuration for N players and K arms, $C^* = (c_i)_{i\in[K]}, c_i \in \{1, \ldots, N\}$. We further define the reward in terms of the probability of correctly identifying the true capacity vector. Formally,

$$\max_{c_1,\ldots,c_K \in [N]} \mathbb{P}\big(C^* = \big[c_1(\mathcal{D}_\tau), \ldots, c_K(\mathcal{D}_\tau)\big]\big),$$

where c_a is a random variable taking values past global history data and output the capacity estimation of arm a. Essentially we can just write

$$\max_{c_1,\ldots,c_K\in[N]} \mathbb{P}\big(C^* = [c_1,\ldots,c_K]\big) = \prod_{a\in[K]} \max_{c\in[N]} \mathbb{P}(C_a^\star = c),$$

Since the capacity hypotheses across arms are independent, we can rewrite this as

$$\max_{\hat{c}_{\tau,1},\ldots,\hat{c}_{\tau,K}\in[N]} \prod_{a\in[K]} \mathbb{P}(c_a^* = \hat{c}_{\tau,a}) = \prod_{a\in[K]} \max_{\hat{c}_{\tau,a}\in[N]} \mathbb{P}(c_a^* = \hat{c}_{\tau,a})\,.$$

Taking logarithms yields

$$\sum_{a\in[K]} \log\Big(\max_{\hat{c}_{\tau,a}\in[N]} \mathbb{P}(c_a^* = \hat{c}_{\tau,a})\Big).$$

Normalizing by K and defining the reward from the inference network as

$$\frac{1}{K}\sum_{a\in[K]} \log\Big(\max_{j\in[N]} \mathbb{P}(c_a^* = \hat{c}_{\tau,a})\Big).$$

B. Experiment Details

B.1. Algorithm

The pseudocode for ICMAPE is provided in Algorithms 1 and 2.

B.2. Baseline Descriptions

Random Policy. In this baseline, each agent selects actions uniformly at random at every time step. A separate inference module monitors the collected data and terminates the process once the estimated probability of correctly identifying all arm capacities exceeds $1-\delta$. This baseline serves as a lower bound on performance without coordinated exploration.

Independent Q-Learning. In this baseline, each agent learns independently using an independent Q-learning framework (Matignon et al., 2012). We refer to this baseline as INDEP in the experiments. Specifically, each agent maintains its own Q-network and updates it using a standard temporal-difference objective (Double DQN). As in our method, a separate inference module predicts the hypothesis, while stopping is controlled by the agents' learned stopping action.

Value Decomposition Networks. We adapt Value Decomposition Networks (VDN) (Sunehag et al., 2017) to our pure exploration setting. The training objective minimizes a temporal-difference loss using the decomposed joint value function $Q_{\text{tot}} = \sum_{i=1}^{N} Q^i$; details are provided in Section A.4. As above, inference and inference reward is handled by a separate module, while termination is controlled by the learned stopping action.

Environment Reward Guided Exploration (Bandit Only). For the multi-armed bandit capacity identification task, we additionally include an environment-reward-guided exploration baseline in the A3P5 setting, where agents are trained using observed bandit rewards rather than inference confidence as the training signal. We include this baseline to demonstrate the designated failure of reward-driven exploration in pure-exploration settings, where maximizing environment rewards does not align with the goal of hypothesis identification.

Thompson Sampling Guided Exploration (Signal Localization Only). Thompson Sampling (TS) is only applicable when the hypothesis space aligns with the action space, as in signal-source localization, where hypotheses correspond to spatial locations that agents can move toward. In the TS baseline, each agent constructs a posterior over hypotheses from its collected data, $\mathcal{D}_t^i$, and samples a hypothesis from this posterior. The sampled hypothesis, together with a predefined action-priority rule, determines the agent's next data-collection action. Agents collect observations until the inference network stops the process. See Section B.8.1 for details.

Algorithm 1 ICMAPE-TD3 TRAINING

Input: prior $\mathcal{P}$, confidence level δ, number of agents N, maximum rollout length T.
Initialize replay buffer $\mathcal{B}$, actors $\{\pi_\theta^i\}_{i=1}^N$, inference network $\mathcal{I}_\psi$, twin critics Q_{ϕ_1}, Q_{ϕ_2}, target networks, and multiplier ζ.
while training is not over **do**
  Sample environment $M \sim \mathcal{P}$ with hypothesis $H^\star$.
  Initialize global trajectory $\mathcal{D}_0$ and local trajectories $\{\mathcal{D}_0^i\}_{i=1}^N$.
  $t \leftarrow 0$.
  repeat
    for each agent $i = 1, \ldots, N$ **do**
      Select action $a_t^i \sim \pi_\theta^i(\cdot \mid \mathcal{D}_t^i)$ with exploration noise.
    end for
    Execute joint action $a_t = (a_t^1, \ldots, a_t^N)$ and observe feedback x_{t+1}.
    Update global trajectory $\mathcal{D}_{t+1}$ and local trajectories $\{\mathcal{D}_{t+1}^i\}_{i=1}^N$.
    Compute posterior $p_\psi(H \mid \mathcal{D}_{t+1})$ and confidence reward $r_{t+1} = \max_H p_\psi(H \mid \mathcal{D}_{t+1})$.
    Set $s_t = \mathbf{1}\{\exists i,\ a_t^i = a_{\text{stop}}\}$ and define cost c_t as in Eq. 4.
    Store $(\mathcal{D}_t, a_t, x_{t+1}, \mathcal{D}_{t+1}, s_t, H^\star)$ in $\mathcal{B}$.
    $t \leftarrow t + 1$.
  until $s_{t-1} = 1$ or $t = T$
  Sample minibatch $B \sim \mathcal{B}$.
  Update inference network $\mathcal{I}_\psi$ using $\mathcal{L}_{\text{inf}}$ in Eq. 3.
  Compute TD3 targets using target actors and $\min_{\ell \in \{1,2\}} Q_{\phi'_\ell}$.
  Update critics Q_{ϕ_1}, Q_{ϕ_2} using $\mathcal{L}_{\text{critic}}$ in Eq. 5.
  if delayed actor update step **then**
    Update actors $\{\pi_\theta^i\}_{i=1}^N$ using $\mathcal{L}_{\text{actor}}$ in Eq. 6.
    Apply soft updates to the target networks.
  end if
  Update multiplier ζ using the batch posterior confidence.
end while

GP Prior Map Guided Sampling (Nitrate Only). For the nitrate concentration experiment, we include an additional baseline motivated by the availability of historical training data. Similar in spirit to the TS baseline in the signal localization setting, we sample 3,000 points from the training set (2001–2018) to construct a GP prior map of the concentration field. Agents then sample independently from this fixed GP prior map without active coordination or inference-driven stopping. This baseline assesses whether a static prior built from historical data suffices for identifying the highest-concentration region, or whether active multi-agent exploration provides meaningful gains.

Algorithm 2 ICMAPE INFERENCE

Input: trained actors $\{\pi_\theta^i\}_{i=1}^N$, trained inference network $\mathcal{I}_\psi$, unknown environment, maximum rollout length T.
Initialize global trajectory $\mathcal{D}_0$ and local trajectories $\{\mathcal{D}_0^i\}_{i=1}^N$.
$t \leftarrow 0$.
repeat
  for each agent $i = 1, \ldots, N$ **do**
    Select action $a_t^i \sim \pi_\theta^i(\cdot \mid \mathcal{D}_t^i)$.
  end for
  Execute joint action $a_t = (a_t^1, \ldots, a_t^N)$ and observe feedback x_{t+1}.
  Update global trajectory $\mathcal{D}_{t+1}$ and local trajectories $\{\mathcal{D}_{t+1}^i\}_{i=1}^N$.
  Set $s_t = \mathbf{1}\{\exists i,\ a_t^i = a_{\text{stop}}\}$.
  $t \leftarrow t + 1$.
until $s_{t-1} = 1$ or $t = T$
Let $\tau = t$ and return

$$\widehat{H}_\tau = \arg\max_{H \in \mathcal{H}} p_\psi(H \mid \mathcal{D}_\tau).$$

B.3. Signal Localization Environment Details

The following are the parameters we used to run the signal localization task and generate Figure 1.

Table 1. Environment parameters for signal source localization.

Parameter	Value
q_i	0.95
d_{min}	10
λ	0.3
σ	0.07
n	25
m	5
Δ_{move}	5

B.4. Maryland Nitrate Concentration Monitoring Details

B.4.1. Data and Spatial Grid

We construct the nitrate monitoring environment from Maryland nitrate concentration measurements obtained from the Water Quality Portal (WQP) (National Water Quality Monitoring Council, 2024). The dataset contains sample-level water-quality observations collected by federal, state, and local monitoring programs, including the sampling location, sampling date, measured characteristic, concentration value, and measurement unit. We use observations from 2001–2023, with years 2001–2018 used for training and years 2019–2023 held out for evaluation. To define a finite sensing domain, we discretize the Maryland region into an $n \times n$ spatial grid. Let $\mathcal{X}$ denote the set of all grid cells, with $|\mathcal{X}| = n^2$. Each cell corresponds to a candidate spatial location at which agents may collect nitrate concentration measurements. Since not all grid cells correspond to valid Maryland monitoring locations, we apply a validity mask and retain only cells that lie within the region of interest and have historical monitoring support. We denote the resulting set of valid sensing locations by $\mathcal{X}_{\text{valid}} \subseteq \mathcal{X}$, with $|\mathcal{X}_{\text{valid}}| = N_{\text{valid}}$. Invalid cells are excluded from the agents' action space and are assigned zero concentration values when constructing spatial fields. For each year, the raw monitoring observations are used to construct an annual nitrate field over the valid grid cells. Specifically, each observation is assigned to its corresponding spatial location, and the resulting irregularly sampled measurements are used as input to the Gaussian-process field-generation procedure described in Appendix B.4.2. This discretized representation allows us to formulate nitrate hotspot identification as a finite pure-exploration and hypothesis testing problem, where the hypothesis set $\mathcal{H}$ is the set of valid grid cells, i.e., $\mathcal{H} = \mathcal{X}_{\text{valid}}$.

B.4.2. GP-Based Field Generation

For each year, we generate spatial nitrate concentration fields using Gaussian process (GP) regression fitted to historical monitoring observations from that year. Since the raw measurements are irregularly distributed across space, the GP provides a smooth spatial interpolation over the discretized Maryland grid. Each generated field defines one latent environment instance for the active-sensing task.

At the beginning of each episode, we first sample a year from the training or evaluation split. We then draw a subset of raw nitrate measurements from that year and fit a GP in log-transformed concentration space. Specifically, if $y(x)$ denotes the nitrate concentration observed at location x, we fit the GP to $\log(1 + y(x))$ to reduce the effect of heavy-tailed concentration values and ensure nonnegative predictions after transformation back to the original scale. The GP is fit using a Matérn kernel with fixed hyperparameters. After fitting, we evaluate the GP posterior mean and standard deviation over all grid cells in $\mathcal{X}$.

Let $\mu_y(x)$ and $\sigma_y(x)$ denote the GP posterior mean and standard deviation in log-concentration space for year y at grid cell x. We construct the latent nitrate field by converting the posterior mean back to the original concentration scale:

$$f_y(x) = \exp(\max\{\mu_y(x), 0\}) - 1.$$

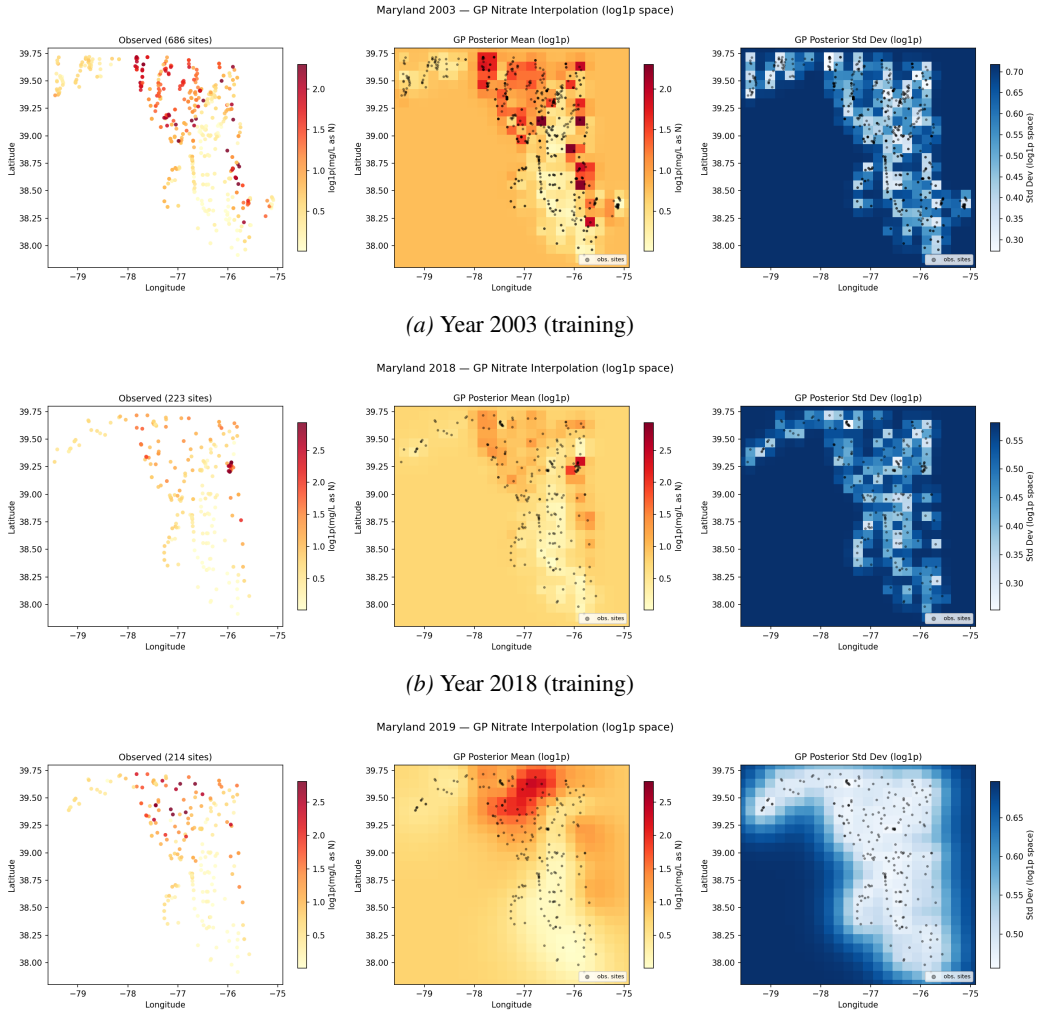
Invalid grid cells are masked out and assigned zero concentration. The true target location for the episode is then defined as the grid cell with the highest concentration,

$$\theta_y^\star = \arg\max_{x \in \mathcal{X}_{\text{valid}}} f_y(x),$$

where $\mathcal{X}_{\text{valid}} \subseteq \mathcal{X}$ denotes the set of valid Maryland sensing locations.

To obtain diverse environment instances, we generate multiple GP-derived fields for each year by repeatedly subsampling the raw observations and refitting the GP. This procedure produces different plausible nitrate concentration maps from the same annual monitoring record, allowing us to evaluate whether an active-sensing policy can generalize across both years and spatial field realizations.

Figure 7 illustrates how the GP-estimated nitrate concentration fields vary across years, showing the average concentration per site for three representative years spanning the training and evaluation periods.



(a) Year 2003 (training)

(b) Year 2018 (training)

(c) Year 2019 (evaluation)

Figure 7. GP-estimated nitrate concentration fields across Maryland for three representative years. Each map is constructed from year-round average concentrations at monitoring sites for that year. Spatial patterns shift notably across years, illustrating the inter-year variability that motivates training on multiple years and evaluating generalization on held-out years.

B.4.3. Multi-Agent Sampling Dynamics

After a GP-derived nitrate field is generated, agents interact with the field through sequential sampling. We consider a team of M agents operating over a finite horizon T. At each timestep t, each agent selects one valid grid cell from $\mathcal{X}_{\text{valid}}$ to sample. The underlying nitrate field f_y is fixed within an episode but is unknown to the agents. Agents can only obtain information about the field through noisy measurements collected at the locations they visit.

When agent i selects location x_t^i, the environment returns noisy nitrate observations generated from the GP posterior at

Year	Raw Data Points	Avg. Var.	Avg. Sigma
2000	10,693	0.265	0.515
2001	10,146	0.287	0.536
2002	11,063	0.249	0.499
2003	11,542	0.264	0.514
2004	11,769	0.421	0.649
2005	10,326	0.238	0.488
2006	8,726	0.289	0.538
2007	8,294	0.282	0.531
2008	9,708	0.214	0.462
2009	8,865	0.219	0.468
2010	5,091	0.362	0.602
2011	5,532	0.319	0.565
2012	2,712	0.391	0.626
2013	3,855	0.317	0.563
2014	5,382	0.265	0.515
2015	5,651	0.298	0.546
2016	5,490	0.320	0.566
2017	1,696	0.533	0.730
2018	5,082	0.276	0.525

Table 2. Per-year data statistics for GP field generation. Avg. Var. denotes the mean GP posterior variance over valid Maryland grid cells in log-transformed concentration space, and Avg. Sigma is its square root.

that cell. In our implementation, each query produces multiple noisy readings. Specifically, observations are sampled in log-concentration space as

$$\tilde{o}^i_{t,j} \sim \mathcal{N}\big(\mu_y(x^i_t), \min\{\sigma_y(x^i_t), \sigma_{\max}\}^2\big)\,, \qquad j = 1, \ldots, n_{\text{obs}},$$

where $\mu_y(x)$ and $\sigma_y(x)$ are the GP posterior mean and standard deviation for year y, and $\sigma_{\max}$ is a cap on the observation noise. The observations are then transformed back to the original concentration scale by

$$o^i_{t,j} = \exp(\tilde{o}^i_{t,j}) - 1.$$

The noise cap prevents highly uncertain grid cells from producing extremely noisy observations, which stabilizes training while preserving spatial uncertainty from the GP posterior.

Each agent maintains its own sampling history,

$$D^i_t = \{(x^i_s, o^i_{s,1}, \ldots, o^i_{s,n_{\text{obs}}})\}_{s=1}^t,$$

and the decentralized actor conditions only on this local history when selecting future actions.

The episode terminates either when the horizon T is reached or when a stopping action is selected. The stopping action is shared at the team level: once any agent chooses to stop, the episode ends for all agents. The final sampling history is then passed to the inference network, which predicts the most likely high-concentration region.

B.4.4. Target Definition and Evaluation

For each GP-generated nitrate field f_y, we define the target as the set of top-k highest-concentration valid grid cells, $\mathcal{H}^\star_y$. We use a top-k target rather than only the single maximum cell because several high-concentration locations may have very similar nitrate levels, making the exact top-1 location sensitive to small GP or observation noise. At the end of each episode, the inference network outputs a confidence score $\widehat{p}(x)$ for each valid cell, and the predicted location is $\widehat{\theta} = \arg\max_{x \in \mathcal{X}_{\text{valid}}} \widehat{p}(x)$. An episode is counted as successful if $\widehat{\theta} \in \mathcal{H}^\star_y$. We report top-$k$ identification accuracy averaged over generated fields and evaluation years, together with the average number of samples used before termination to measure sample efficiency.

Table 3 summarizes the main variables and parameters used in the nitrate monitoring enviroment.

Symbol / Parameter	Value	Description
$\mathcal{X}$	$n \times n$ grid	Discretized Maryland sensing domain
n	20	Grid resolution along each spatial dimension
$\mathcal{X}_{\text{valid}}$	valid grid cells	Cells inside the region of interest with historical monitoring support
N_{valid}	$\lvert\mathcal{X}_{\text{valid}}\rvert$	Number of valid sensing locations
M	6	Number of agents
T	40	Maximum episode horizon
k	5	Number of high-concentration target cells
n_{obs}	5	Number of noisy readings returned per query
$\mathcal{Y}_{\text{train}}$	2001–2018	Training years
$\mathcal{Y}_{\text{test}}$	2019–2023	Held-out evaluation years
N_{fit}	100	Number of raw monitoring observations used to fit each training GP field
f_y	GP-derived field	Latent nitrate concentration field for year y
$\mathcal{H}_y^\star$	top-k cells	True high-concentration hotspot set for year y
$\sigma_{\max}$	0.5	Cap on GP posterior standard deviation in log space
N_{GP}	100	Number of GP-generated maps sampled per year for evaluation
N_{points}	1000	Number of spatial points sampled from each GP-generated map for evaluation
N_{train}	400,000	Maximum number of training steps

Table 3. Main variables and parameters used in the Maryland nitrate concentration monitoring environment.

B.5. Model Architecture

B.5.1. Capacity Identification Model Architecture

For the bandit capacity identification task, the model operates on sequential trajectory data collected by the agents.. The architecture is based entirely on sequence encoders that process the history of observed actions and rewards. We use the LSTM-based encoder with conditional pooling described in Section B.7 to summarize the trajectory up to the current timestep.

Actor Network. The actor network parameterizes the decentralized exploration policy for each agent. Given an agent's local trajectory $\mathcal{D}_t^i$, the input sequence is first mapped into an embedding space through a linear layer followed by a GELU activation. The embedded sequence is then passed through the LSTM sequence encoder described in Section B.7.

Let $\widetilde{h}_t^i$ denote the pooled trajectory representation for agent i. The actor maps this representation to logits over the action space with a linear layer. where A is the number of bandit arms and the additional action corresponds to the stopping action a_{stop}. During training, we use Gumbel–Softmax sampling with temperature τ to obtain differentiable action representations for TD3-style policy optimization. At execution time, each agent acts using only its own local trajectory $\mathcal{D}_t^i$, preserving decentralized execution.

Critic Network. The critic is implemented as a twin LSTM-based critic. The global trajectory $\mathcal{D}_t$ is embedded with a linear layer followed by a GELU activation, processed by an LSTM encoder, and pooled using the attention-based pooling layer from Section B.7. The critic also conditions on the joint action through a masked one-hot encoding: for each queried agent i, the agent's own action is masked out, leaving only the actions of the other agents, denoted by $\widetilde{a}_t^{-i}$. The pooled trajectory representation $\widetilde{h}_t$ and the masked action representation are concatenated and passed through a two-layer MLP to produce the Q-value estimate:

$$Q_{\phi_\ell}^i(\mathcal{D}_t, a_t^{-i}) = f_{\phi_\ell}\big([\widetilde{h}_t; \widetilde{a}_t^{-i}]\big), \qquad \ell \in \{1, 2\}.$$

The two critics use separate LSTM encoders and separate output heads, and TD3 targets are computed using the minimum of the two target critic values.

Inference Network. The inference network predicts the underlying capacity vector from the global trajectory $\mathcal{D}_t$. It uses the same LSTM-based sequence encoder described in Section B.7. The input trajectory is first embedded through a linear layer followed by a GELU activation. The embedded sequence is then processed by the LSTM encoder, and attention-based pooling with a timestep mask is used to obtain a fixed-dimensional trajectory representation $\widetilde{h}_t$.

The pooled representation is mapped through a linear output layer to produce logits over possible capacities for each arm:

$$o_\psi(\mathcal{D}_t) = W_{\text{out}}\widetilde{h}_t + b_{\text{out}}, \qquad o_\psi(\mathcal{D}_t) \in \mathbb{R}^{A \times N},$$

where A is the number of arms and N is the number of players. During training, the inference network is supervised using the true capacity vector $C^\star$.

B.5.2. Signal Localization Model Architecture

The signal localization model uses a different architecture from the bandit model because the input contains both spatial observations and temporal location histories. Specifically, each agent observes a local belief map on an $n \times n$ grid, while the model also receives the location history of all agents over time. We process these two sources of information separately and then fuse them to produce each agent's action distribution.

Actor Network. The actor is implemented as a per-agent CNN–LSTM architecture. For each agent i, the local belief map is first processed by a player-specific CNN encoder, producing a spatial embedding $z^i_{\text{grid}} \in \mathbb{R}^d$. In parallel, the full location history of all agents is reshaped into a sequence of joint locations and passed through a player-specific LSTM, producing a temporal location embedding $z^i_{\text{loc}} \in \mathbb{R}^d$ at the current timestep using LSTM with conditional pooling mentioned in Section B.7.

The two embeddings are then concatenated and passed through a player-specific fusion module, implemented as a linear projection followed by a GELU activation. Finally, a player-specific output layer maps z^i_{fuse} to logits over $A + 1$ actions, where the additional action corresponds to the stopping action. The policy distribution is obtained by applying a softmax to these logits. During TD3 training, we use Gumbel–Softmax samples to obtain differentiable action representations for the actor update.

Critic Network. The critic for the signal localization environment is implemented as a twin centralized critic. Unlike the actor, which conditions only on each agent's local trajectory, the critic has access to global trajectory information during training. Specifically, it takes as input the belief maps for all agents, the joint location history, and the joint action.

For each critic head, the stacked belief maps are first processed by a CNN encoder. Since the critic observes the belief maps of all agents, the input is represented as a multi-channel grid, where the channels correspond to the agents' belief maps. The CNN encoder produces a spatial embedding $z_{\text{grid}} \in \mathbb{R}^d$. In parallel, the location history of all agents is reshaped into a sequence of joint locations and passed through an LSTM, producing a temporal embedding $z_{\text{loc}} \in \mathbb{R}^d$.

To construct an agent-specific critic input, the spatial and temporal embeddings are shared across agents and concatenated with a one-hot encoding of the queried agent identity. The critic also receives an action feature vector encoding the actions of the other agents. Following our bandit critic design, we mask out the queried agent's own action so that the critic for agent i conditions on a^{-i} rather than directly on a^i.

The resulting critic representation is

$$z^i_{\text{critic}} = \left[z_{\text{grid}}; z_{\text{loc}}; e_i; a^{-i}\right],$$

where e_i is the one-hot encoding of agent i. This representation is passed through a feed-forward fusion layer followed by a ReLU activation, and then through a linear output layer to produce the Q-value estimate for each agent.

To reduce overestimation bias, we use two independent critic heads, as in TD3. Each critic head has its own CNN encoder, location LSTM, fusion layer, and output layer. Thus, the critic outputs

$$\{Q^i_{\phi_1}, Q^i_{\phi_2}\}_{i=1}^N,$$

which are used to compute TD3 targets by taking the minimum over the two critic estimates.

Inference Network. The inference network for signal localization predicts the hidden source location from the aggregated grid representation. Unlike the actor and critic, which use both spatial observations and location histories, the inference network is implemented as a fully convolutional network over the grid input. Given a multi-channel grid $G_t \in \mathbb{R}^{C \times n \times n}$ constructed from the global trajectory $\mathcal{D}_t$, the network outputs logits over all spatial locations:

$$\ell_\psi(G_t) \in \mathbb{R}^{n \times n}.$$

The architecture consists of two 3×3 convolutional layers with ReLU activations, followed by a 1×1 convolution that maps the hidden feature maps to a single logit per grid cell.

The posterior distribution over candidate source locations is obtained by applying a softmax over the flattened grid:

$$p_\psi(H = (u, v) \mid \mathcal{D}_t) = \frac{\exp(\ell_\psi(G_t)_{u,v})}{\sum_{u'=1}^{n} \sum_{v'=1}^{n} \exp(\ell_\psi(G_t)_{u',v'})}.$$

The final estimate is given by the maximum a posteriori prediction:

$$\widehat{H}_t = \arg\max_{(u,v)} p_\psi(H = (u, v) \mid \mathcal{D}_t).$$

During training, the inference network is supervised using the true source location $H^\star$, and its posterior confidence is used to define the learning signal for the exploration policy.

B.6. Nitrate Concentration Architecture

For the nitrate monitoring task, we use a CNN-LSTM architecture for both the centralized critic and the inference network. The main difficulty is that useful information is naturally organized by spatial cell, while the raw sampling history is organized as a sequence of agent-time observations. For example, observations from the same cell may appear at different timesteps and may be collected by different agents. A standard LSTM or Transformer over the raw sequence must learn to associate these scattered observations with the same spatial cell and aggregate them across both time and agents, which can be inefficient and unstable. The CNN-LSTM architecture addresses this by first constructing cell-level spatial summaries and then using a convolutional encoder to extract spatial features.

B.6.1. CNN-LSTM Architecture

To address this, we first convert the joint sampling history into grid-level summary statistics. For each grid cell, we aggregate all observations collected at that location across agents and timesteps, including statistics such as the normalized mean, standard deviation, maximum observed value, sampling count, and whether the cell has been sampled. This produces a structured $n \times n$ spatial representation of the team's collected information. A CNN encoder is then applied to this grid tensor to extract spatial features that capture local concentration patterns and sampling coverage.

In parallel, an LSTM encoder processes each agent's action history to summarize temporal sampling behavior. The CNN-based spatial representation and the LSTM-based history representation are then combined to form a global representation of the current sampling state. In the centralized critic, this representation is used to output action-value estimates for each agent. In the inference network, the same type of CNN-LSTM backbone is used to predict confidence scores over the valid grid cells, indicating which locations are likely to belong to the high-concentration target set.

We use the CNN-LSTM architecture for the critic and inference network, where centralized training permits access to the full joint sampling history. During execution, the actor remains decentralized and conditions only on each agent's local sampling history.

B.7. Recurrent Sequence Encoder with Conditioned Pooling

Each network in our framework takes as input a trajectory $x_{1:t}$ such as observations or state–action sequences. At a given timestep t, the sequence $x_{1:t}$ is first mapped into an embedding space using a linear layer, and then processed by a recurrent encoder:

$$h_{1:t} = \text{LSTM}(\text{Embed}(x_{1:t})).$$

Depending on the environment, this encoder can either be shared across agents when agents are homogeneous; or instantiated separately when agents have different observation spaces. In either case, the overall architecture remains the same. To summarize the sequence of hidden state into a fixed-dimensional representation, we use a *conditioned temporal pooling* operator, which lets the model adaptively focus on relevant parts of the trajectory.

Conditioned Temporal Pooling We define the pooled representation at time t as $c_t = \sum_{s=1}^{t} \alpha_s h_s$, where the weights α_s are computed using a learned query-key interaction $\alpha_s \propto \exp\big(\frac{\langle q_{\text{in}}, W_k h_s \rangle}{\sqrt{d}}\big)$. Here, q_{in} is a learned query vector and W_k is a linear projection. In practice, sequences are padded to a fixed horizon, and we apply a mask during pooling so that the representation at time t only uses information from steps 1 through t, ignoring padded entries and preventing leakage from future timesteps. The resulting context vector c_t is then passed through a gating mechanism, $\tilde{h}_t = c_t \odot \sigma(W_g q_{\text{out}})$, where

q_{out} is another learned query vector and $\sigma(\cdot)$ is a nonlinear activation function. This conditioned pooling allows the model to emphasize the most informative parts of the trajectory, rather than relying only on the final hidden state or uniformly averaging over timesteps. Each network (actor, critic, and inference) uses its own instance of the sequence encoder described above, sharing only the same architecture but not parameters.

B.8. Posterior Thompson Sampling Baseline

B.8.1. Agent-wise Posterior Construction

We describe the posterior used by each agent in the posterior Thompson-sampling baseline. Each agent constructs a belief using its own observations and the observed trajectories of the other agents. Let $s^\star \in [n] \times [n]$ denote the true source location, and let $d^\star \in [n] \times [n]$ denote the distractor location. For agent i, let u_t^i denote its position at time t, and let $\mathcal{V}_t^i = \{u_\tau^i : \tau \le t\}$ denote the cells visited by agent i up to time t.

Coverage likelihood. We first define the negative-evidence contribution induced by an agent's trajectory. For an agent $j \in \{1, 2\}$ that can miss the true source with probability δ_j, observing that agent j visited a cell u_τ^j without identifying the source downweights the hypothesis that $s^\star = u_\tau^j$. We write this coverage likelihood as

$$\ell_j(s, u_\tau^j) = \begin{cases} \delta_j, & u_\tau^j = s, \\ 1, & u_\tau^j \neq s. \end{cases}$$

The cumulative coverage likelihood from a set of agents $\mathcal{J}$ is

$$L_{\mathcal{J}}(s) = \prod_{j \in \mathcal{J}} \prod_{\tau \le t} \ell_j(s, u_\tau^j).$$

Posteriors for agents 1 and 2. Agents 1 and 2 are binary detectors of the true source. For $i \in \{1, 2\}$, agent i maintains a posterior over the true source location,

$$p_t^i(s) = \mathbb{P}(s^\star = s \mid \mathcal{D}_t^i, \mathcal{V}_t^{-i}),$$

where $\mathcal{D}_t^i$ is agent i's local binary observation history and $\mathcal{V}_t^{-i}$ denotes the observed trajectories of the other agents. The prior is initialized uniformly:

$$p_0^i(s) = \frac{1}{n^2}, \qquad s \in [n] \times [n].$$

Let $y_t^i \in \{0, 1\}$ denote the binary observation of agent i. If agent i visits the true source cell, it misses the target with probability δ_i. Thus, if $u_t^i = s$,

$$P(y_t^i = 1 \mid s) = 1 - \delta_i, \qquad P(y_t^i = 0 \mid s) = \delta_i.$$

If $u_t^i \neq s$, we assume a small false-positive probability ϵ_i (where $\epsilon_{1,2} = 0$):

$$P(y_t^i = 1 \mid s) = \epsilon_i, \qquad P(y_t^i = 0 \mid s) = 1 - \epsilon_i.$$

Agent i also uses the trajectories of the other binary detector and agent 3 as negative evidence. Therefore, for $i \in \{1, 2\}$, define

$$\mathcal{J}_i = \{1, 2, 3\} \setminus \{i\}.$$

The posterior used by agent i is

$$p_t^i(s) = \frac{p_0^i(s) \left[\prod_{\tau \le t} P(y_\tau^i \mid s)\right] L_{\mathcal{J}_i}(s)}{\sum_{s'} p_0^i(s') \left[\prod_{\tau \le t} P(y_\tau^i \mid s')\right] L_{\mathcal{J}_i}(s')}.$$

Equivalently, the recursive update is

$$p_t^i(s) \propto p_{t-1}^i(s) P(y_t^i \mid s) \prod_{j \in \mathcal{J}_i} \ell_j(s, u_t^j), \qquad i \in \{1, 2\}.$$

This posterior downweights cells that were visited by other agents without source identification, while still using agent i's own binary observations as direct evidence.

Posterior for agent 3. Agent 3 observes a continuous signal field affected by both the true source and an unknown distractor. Therefore, agent 3 maintains a joint posterior over source-distractor pairs:

$$p_t^3(s,d) = \mathbb{P}(s^\star = s, d^\star = d \mid \mathcal{D}_t^3, \mathcal{V}_t^1, \mathcal{V}_t^2).$$

The prior is initialized uniformly over valid source-distractor pairs:

$$p_0^3(s,d) \propto 1, \qquad s \neq d,$$

and

$$p_0^3(s,d) = 0 \qquad \text{if } s = d.$$

For a candidate pair (s,d), the expected signal at cell u is

$$m_{s,d}(u) = \max\left\{\exp(-\lambda \operatorname{dist}(u,s)), 0.7\exp(-\lambda \operatorname{dist}(u,d))\right\}.$$

Here, the first term corresponds to the true source and the second term corresponds to the distractor with intensity 0.7. Let W_t^3 denote the local observation window of agent 3 at time t. For each cell $u \in W_t^3$, agent 3 observes a continuous value $y_t^3(u)$, modeled as

$$y_t^3(u) \mid s,d \sim \mathcal{N}\big(m_{s,d}(u), \sigma^2\big).$$

Assuming conditionally independent observation noise across cells in the local window, the likelihood of agent 3's observation is

$$P(y_t^3 \mid s,d) = \prod_{u \in W_t^3} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\big(y_t^3(u) - m_{s,d}(u)\big)^2}{2\sigma^2}\right).$$

Agent 3 also uses the observed trajectories of agents 1 and 2 as negative evidence. Since agents 1 and 2 miss the true source with probabilities δ_1 and δ_2, respectively, their coverage likelihood is

$$L_{\{1,2\}}(s) = \prod_{j=1}^{2} \prod_{\tau \le t} \ell_j(s, u_\tau^j).$$

Combining agent 3's continuous signal likelihood with this coverage information, the posterior for agent 3 is

$$p_t^3(s,d) = \frac{p_0^3(s,d)\left[\prod_{\tau \le t} P(y_\tau^3 \mid s,d)\right] L_{\{1,2\}}(s)}{\sum_{s'}\sum_{d'} p_0^3(s',d')\left[\prod_{\tau \le t} P(y_\tau^3 \mid s',d')\right] L_{\{1,2\}}(s')}.$$

Equivalently, the recursive update is

$$p_t^3(s,d) \propto p_{t-1}^3(s,d) P(y_t^3 \mid s,d) \ell_1(s,u_t^1)\ell_2(s,u_t^2).$$

Since the final goal is to identify the true source rather than the distractor, agent 3 marginalizes over the distractor:

$$p_t^3(s) = \sum_d p_t^3(s,d).$$

Posterior sampling. After updating its posterior, each agent samples a candidate true source location. For agents 1 and 2,

$$\tilde{s}_t^i \sim p_t^i(s), \qquad i \in \{1,2\}.$$

For agent 3,

$$\tilde{s}_t^3 \sim p_t^3(s), \qquad p_t^3(s) = \sum_d p_t^3(s,d).$$

The sampled location $\tilde{s}_t^i$ is then used to determine the next movement action of agent i.

B.8.2. Deterministic Movement Rule After Posterior Sampling

Because agents cannot directly query arbitrary cells, each agent moves one step toward its sampled source using a fixed priority order.

Agent 1 uses the order

$$(\text{down}, \text{left}, \text{up}, \text{right}),$$

agent 2 uses

$$(\text{right}, \text{down}, \text{up}, \text{left}),$$

and agent 3 uses

$$(\text{right}, \text{up}, \text{left}, \text{down}).$$

Let $\text{Move}(p, a)$ be the position obtained by taking action a from position p, with boundary clipping. Agent i selects the first action a in its priority list such that

$$\text{dist}(\text{Move}(p_t^i, a), \tilde{s}_t^i) < \text{dist}(p_t^i, \tilde{s}_t^i).$$

If no action decreases the distance, the agent stays. The different priority orders break symmetry and encourage different movement patterns across agents.

B.8.3. Stopping Rule

Action selection is decentralized and posterior-based, but stopping uses the same aggregated inference module as the main method. At each time t, the global trajectory $\mathcal{D}_t$ is passed to the inference module, which outputs a posterior $q_\psi(s \mid \mathcal{D}_t)$ over source locations. The inferred source is $\hat{s}_t = \arg\max_s q_\psi(s \mid \mathcal{D}_t)$, and the confidence is $r_t = \max_s q_\psi(s \mid \mathcal{D}_t)$.

The baseline stops at $\tau = \inf\{t : \max_s q_\psi(s \mid \mathcal{D}_t) \geq 1 - \delta\}$ and outputs $\hat{s}_\tau = \arg\max_s q_\psi(s \mid \mathcal{D}_\tau)$. Thus, this baseline differs from the learned policy in the action-selection rule.

B.9. Implementation Details

Seed Option All experiments are conducted using three randomly selected seeds: 215, 219, and 321.

Software dependencies. All experiments are implemented in Python. We use PyTorch (Paszke et al., 2019) for neural network training, Gymnasium (Towers et al., 2024) for environment implementation, Hydra (Yadan, 2019) for configuration management, and Weights & Biases (Biewald, 2020) for experiment tracking and visualization. Data processing and analysis are performed using NumPy (Harris et al., 2020), Pandas (McKinney, 2010), Matplotlib (Hunter, 2007), and Seaborn (Waskom, 2021). The main package versions used in our experiments are:

$$\texttt{numpy} = 2.2.6, \quad \texttt{pandas} = 2.3.1, \quad \texttt{gymnasium} \geq 1.0.0, \quad \texttt{torch} = 2.7.1, \quad \texttt{matplotlib} = 3.10.$$

Hardware. All experiments are run on NVIDIA L40S GPUs with 48GB of GPU memory. Capacity identification experiments take approximately 60 hours to complete, while signal localization experiments finish within approximately 15 hours. The observed GPU utilization during training is typically around 40–50%.

C. Additional Experiment Analysis Detailes

C.1. Capacity Identification.

We further analyze the stopping behavior of different methods using survival plots for the bandit environments (Figure 8). In both the 3-arm 5-player (A3P5) and 5-arm 3-player (A5P3) settings, the TD3-based architecture exhibits a clear exponential decay in survival probability, indicating that the model learns to terminate exploration efficiently once sufficient confidence is achieved. The plots report the average survival probability across three random seeds for each method.

C.1.1. Model Robustness.

We evaluate the robustness of our framework under different prior distributions over the environment parameters. In the default setting, both arm capacities and arm means are sampled uniformly. To test robustness with respect to the arm-mean

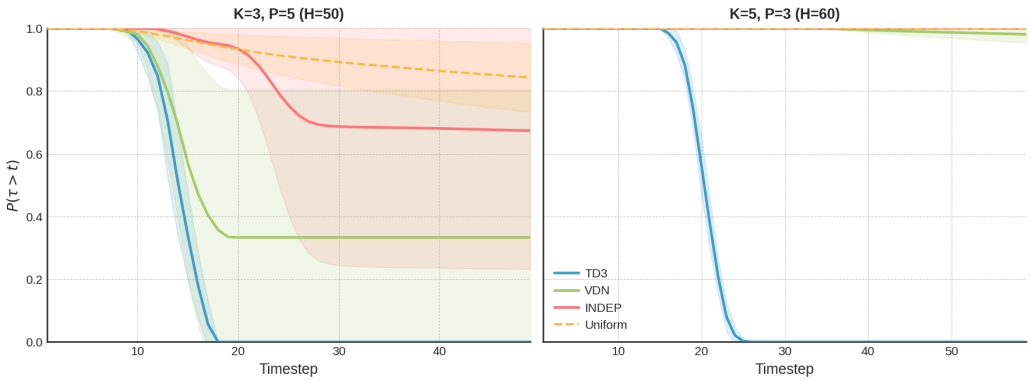


Figure 8. Survival plots for the bandit environments. The survival probability measures the fraction of runs that have not yet selected the stopping action at each timestep. Curves are averaged over three random seeds for each method.

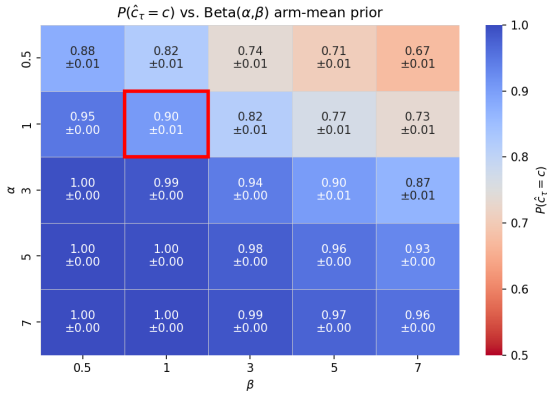
prior, we replace the uniform prior with a scaled Beta distribution parameterized by (α, β) and supported on $[0.3, 1]$. When $\alpha > \beta$, the distribution places more mass on larger arm means, resulting in environments with more informative feedback and consequently higher inference accuracy.

Figure 9 reports the average inference accuracy and average stopping time under different Beta priors. Row (a) corresponds to the A3P5 setting, while Row (b) corresponds to the A5P3 setting. In each row, the left panel shows inference accuracy and the right panel shows the average stopping time.

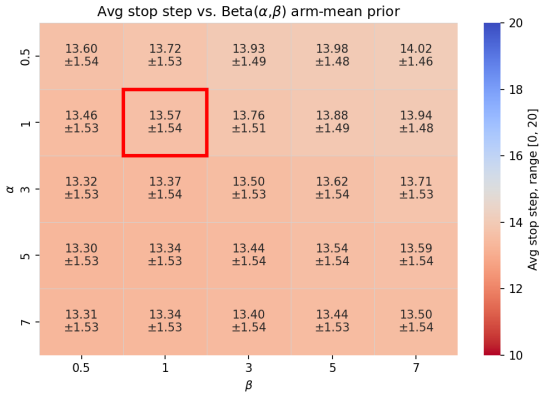
We additionally study robustness with respect to the arm-capacity prior. Instead of sampling capacities uniformly, we generate a probability vector over capacities using a symmetric Dirichlet distribution,

$$p \sim \mathrm{Dir}(\theta \mathbf{1}_K),$$

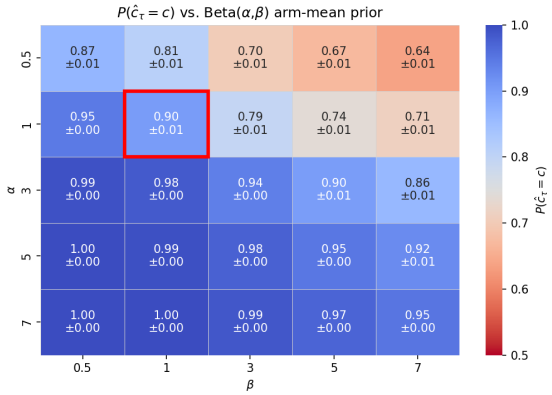
and then sample each arm capacity according to the resulting distribution. We consider $\theta \in \{0.1, 0.5, 1, 5, 10\}$, where smaller values of θ produce more skewed capacity distributions, while larger values approach a uniform distribution. Figure 10 summarizes the resulting inference accuracy under different Dirichlet priors.



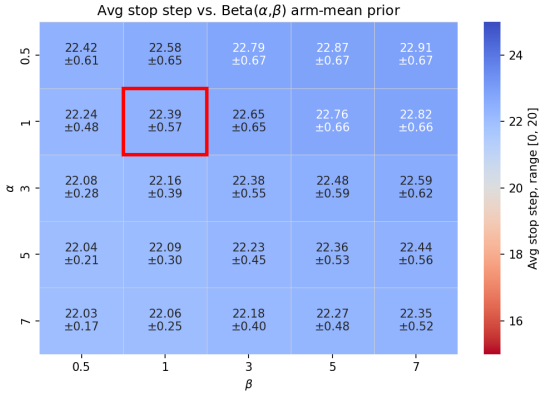
(a) Inference accuracy in A3P5.



(b) Average stopping time in A3P5.

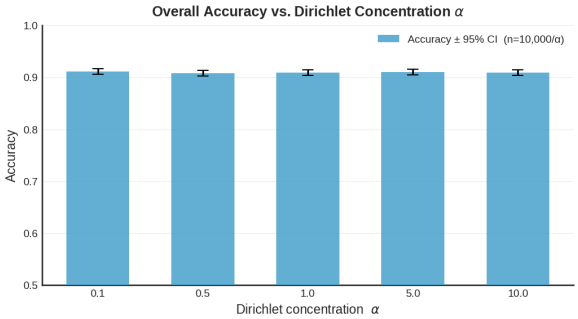


(c) Inference accuracy in A5P3.



(d) Average stopping time in A5P3.

Figure 9. Robustness analysis under different scaled Beta priors over arm means. Panels (a) and (c) show inference accuracy, while panels (b) and (d) show average stopping time. The top row corresponds to the A3P5 setting and the bottom row corresponds to the A5P3 setting.



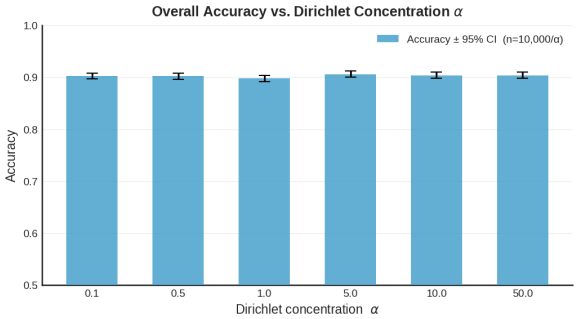


Figure 10. Robustness analysis under different Dirichlet priors over arm capacities. Left: A3P5 setting. Right: A5P3 setting.

C.1.2. Coordination Strategy

Table 4. Most common joint exploration action at each step across 10,000 sampled environments (A3P5) under the greedy ICMAPE-TD3 policy. At step t, only environments still active (stopping time $\geq t$) are counted (n_{active}). The joint action $(a^{(1)}, a^{(2)}, a^{(3)}, a^{(4)}, a^{(5)})$ lists the arm index (0–2) pulled by each player in the modal joint action. Steps beyond t=15 are omitted as all environments have stopped.

t	Joint action	Count	n_{active}	%
0	$(2,2,2,2,2)$	10,000	10,000	100.0
1	$(2,2,2,2,2)$	10,000	10,000	100.0
2	$(2,2,2,2,2)$	10,000	10,000	100.0
3	$(2,0,2,2,2)$	10,000	10,000	100.0
4	$(2,0,0,2,2)$	10,000	10,000	100.0
5	$(0,0,0,0,0)$	10,000	10,000	100.0
6	$(0,0,0,0,1)$	10,000	10,000	100.0
7	$(1,0,0,0,1)$	10,000	10,000	100.0
8	$(1,0,0,1,1)$	10,000	10,000	100.0
9	$(1,1,1,1,1)$	10,000	10,000	100.0
10	$(1,2,1,1,1)$	9,031	10,000	90.3
11	$(1,2,2,1,1)$	6,279	9,031	69.5
12	$(1,2,2,1,1)$	4,510	6,279	71.8
13	$(1,2,2,2,1)$	4,236	5,752	73.6
14	$(0,0,0,0,0)$	3,872	4,236	91.4
15	$(0,0,0,0,0)$	364	364	100.0

Table 5. Frequency of joint stop-action combinations across 10,000 sampled environments (A3P5). A combination is recorded when one or more players simultaneously select the stop action at the terminal step. *% vol.* is the fraction of voluntary-stop episodes; *% all* is the fraction of all 10,000 environments.

Combination	Count	% vol. stops	% all envs
1 player stopping simultaneously			
$\{P_5\}$	1,778	17.8	17.8
$\{P_4\}$	1,056	10.6	10.6
$\{P_1\}$	947	9.5	9.5
$\{P_3\}$	10	0.1	0.1
2 players stopping simultaneously			
$\{P_1, P_5\}$	2,723	27.2	27.2
$\{P_4, P_5\}$	813	8.1	8.1
$\{P_1, P_4\}$	382	3.8	3.8
$\{P_3, P_5\}$	18	0.2	0.2
$\{P_2, P_4\}$	14	0.1	0.1
$\{P_1, P_3\}$	2	0.0	0.0
$\{P_3, P_4\}$	1	0.0	0.0
3 players stopping simultaneously			
$\{P_1, P_4, P_5\}$	1,806	18.1	18.1
$\{P_1, P_3, P_5\}$	125	1.2	1.2
$\{P_2, P_4, P_5\}$	25	0.2	0.2
$\{P_3, P_4, P_5\}$	3	0.0	0.0
$\{P_1, P_2, P_4\}$	1	0.0	0.0
$\{P_1, P_3, P_4\}$	1	0.0	0.0
4 players stopping simultaneously			
$\{P_1, P_3, P_4, P_5\}$	257	2.6	2.6
$\{P_1, P_2, P_4, P_5\}$	5	0.1	0.1
$\{P_2, P_3, P_4, P_5\}$	1	0.0	0.0
$\{P_1, P_2, P_3, P_5\}$	1	0.0	0.0
5 players stopping simultaneously			
$\{P_1, P_2, P_3, P_4, P_5\}$	31	0.3	0.3

C.2. Signal Source Localization

We provide additional results for the signal source localization environment, complementing the training dynamics and final performance results reported in the main text.

We further analyze the stopping behavior of different methods in the signal source localization environment using survival plots (Figure 11).

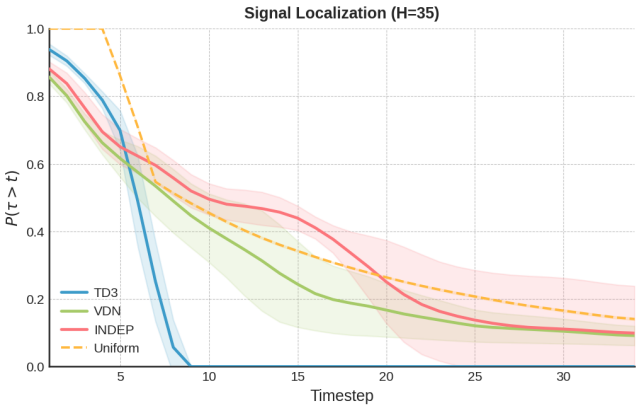


Figure 11. Survival plots for the signal source localization environment. The survival probability measures the fraction of runs that have not yet selected the stopping action at each timestep. Curves are averaged over three random seeds for each method.

Model Robustness. We also evaluate model robustness in the signal source localization environment. In the default setting, the minimum Manhattan distance between the true signal source and the distractor is set to 10, and source distractor pairs are sampled uniformly at random. To study the effect of source distractor separation, we fix the Manhattan distance between the two points to values ranging from 10 to 48. For each distance value, we sample 10,000 instances and report the resulting inference accuracy in Figure 12.

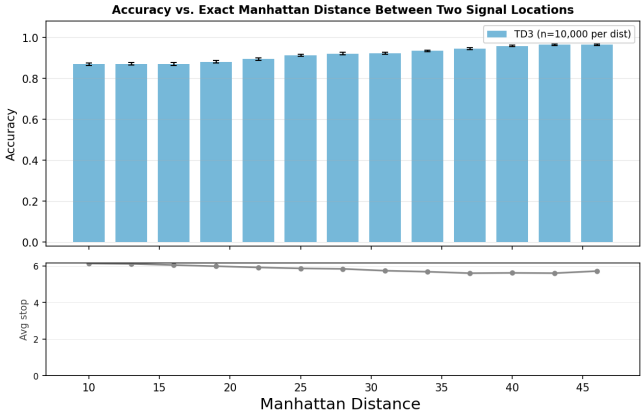


Figure 12. Inference accuracy as a function of the Manhattan distance between the true signal source and the distractor. Larger distances generally make the two sources easier to distinguish. Results are computed over 10,000 sampled instances for each distance value.

Similar to the robustness analysis over arm means in the capacity identification environment, we also test robustness with respect to the distractor intensity distribution. Specifically, we sample the distractor intensity from a scaled Beta distribution supported on $[0, 0.7]$ with different (α, β) parameters. Figure 13 reports the inference accuracy under these different distractor-intensity priors.

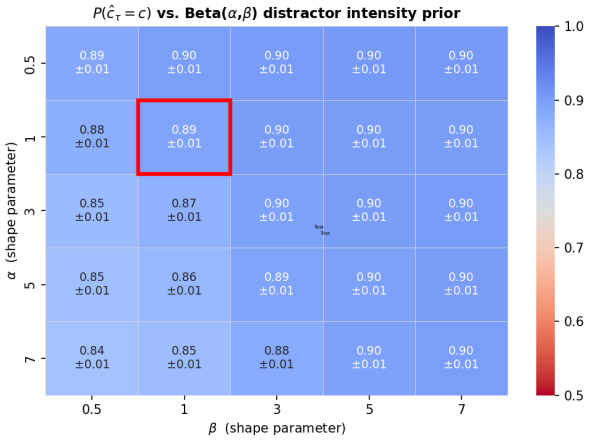


Figure 13. Robustness analysis under different scaled Beta priors over distractor intensity in the signal source localization environment. The heatmap reports inference accuracy for different choices of (α, β), with distractor intensities scaled to lie in $[0, 0.7]$.

C.2.1. Coordination Strategy

Table 6. Most common action for each agent at each step across active environments under the greedy ICMAPE-TD3 policy in the signal localization (10,000 samples). Each cell in the table reports the agent action and its frequency among the n_{active} environments still running at that step. Action labels: **up** = up **dn** = down, **lf** = left, **rt** = right; **STOP** = stop action. Agent 1 and agent 2 maintains a fully deterministic trajectory throughout

t	Agent 1	Agent 2	Agent 3	n_{active}
0	dn (100.0%)	dn (100.0%)	rt (99.9%)	10,000
1	dn (100.0%)	dn (100.0%)	rt (94.6%)	9,994
2	rt (100.0%)	dn (100.0%)	rt (92.6%)	9,459
3	up (100.0%)	dn (100.0%)	up (95.2%)	8,759
4	up (100.0%)	dn (100.0%)	up (93.0%)	8,338
5	up (100.0%)	dn (100.0%)	up (87.5%)	7,755
6	rt (100.0%)	dn (100.0%)	lf (86.0%)	6,783
7	rt (100.0%)	**STOP** (100.0%)	up (87.3%)	5,835

Table 7. Frequency of joint stop-action combinations across 10,000 sampled environments in the robot wildfire task (Signal Localization). *% vol.* is the fraction of voluntary-stop episodes; *% all* is the fraction of all 10,000 environments.

Combination	Count	% vol. stops	% all envs
1 robot stopping simultaneously			
$\{P_2\}$	5,095	50.9	50.9
$\{P_3\}$	4,165	41.6	41.6
2 robots stopping simultaneously			
$\{P_2, P_3\}$	740	7.4	7.4

D. Limitations and Broader Impact

Several limitations suggest directions for future work. Our current framework is trained using an off-policy actor–critic method. While this provides sample reuse and stable value learning, future work could explore on-policy alternatives such as PPO (Schulman et al., 2017) to better understand how different reinforcement learning objectives affect decentralized information acquisition and coordination strategy in our setting. Another limitation is that our experiments focus on fixed hypothesis spaces, where the underlying hypothesis remains constant throughout an episode. A possible direction is to study more complex environments in which the hypothesis evolves over time, requiring agents to continually adapt their exploration strategies. More broadly, our work studies hypothesis identification from a prior distribution over environments. Future research could extend this setting toward more general forms of environment understanding, where agents are not

restricted to identifying a pre-specified hypothesis, but instead learn richer representations of the sampled environment that support a wider range of downstream inference tasks.