*HHAI 2026*
*M. Alimardani et al. (Eds.)*

*doi:10.3233/FAIA260495*

# RecBayes: Scalable *Ad Hoc* Teamwork Without Privileged Information

João G. RIBEIRO [a], Yaniv OREN [b], Alberto SARDINHA [c], Matthijs SPAAN [b], Francisco S. MELO [d]

[a] *Utrecht University*
[b] *Delft University of Technology*
[c] *Pontifical Catholic University of Rio de Janeiro*
[d] *Técnico Lisboa, University of Lisbon*

**Abstract.** Hybrid intelligence scenarios — where artificial agents must collaborate with pre-existing human teams in the real world — require agents capable of joining those teams on-the-fly, without prior coordination or communication, and without being able to directly observe the internal states or intended actions of their human teammates. *Ad hoc* teamwork formalises precisely this challenge, yet current approaches remain constrained by requirements that are fundamentally incompatible with human-AI teaming: either they require access to privileged information such as the full environment state or the real-time actions of teammates [Barrett et al., 2017, Gu et al., 2021, Rahman et al., 2023], or they are limited to environments small enough to be tabularly modelled as partially observable Markov decision processes [Ribeiro et al., 2023a]. In this paper we introduce RecBayes, a recurrent Bayesian approach to *ad hoc* teamwork that lifts both constraints simultaneously. RecBayes identifies teammates and the tasks they are performing using only partial observations of the environment, never requiring access to teammate actions or environment states at any stage, and scales to domains with arbitrarily large state and observation spaces. We demonstrate its effectiveness in two benchmark multi-agent domains with up to $1M$ states and $2^{125}$ possible observations, showing that RecBayes correctly identifies teams and tasks from partial observations alone, and does so efficiently enough to assist those teams in completing their objectives. These properties make RecBayes a principled step toward *ad hoc* agents deployable in real hybrid human-AI teams.

**Keywords.** Ad Hoc Teamwork, Partial Observability, Multi-Agent Systems, Human-AI Teaming, Reinforcement Learning

## 1. Introduction

A central aspiration of hybrid intelligence research is to build artificial agents that can collaborate seamlessly with human teams as active participants that join existing groups, understand what those groups are trying to accomplish and contribute accordingly [Akata et al., 2020, Dafoe et al., 2020]. Achieving this requires agents that can be deployed on-the-fly into pre-existing teams, without any opportunity for pre-coordination or pre-

communication, and without being able to directly observe the internal states or intended actions of their human partners. This is precisely the setting studied in the *ad hoc* teamwork literature [Stone et al., 2010], which frames the problem as one of simultaneous identification and assistance: an agent is dropped into an ongoing team interaction, must infer the team's composition and goal from whatever partial evidence it can observe, and must act to help the team accordingly.

Despite its natural alignment with hybrid intelligence goals, existing *ad hoc* teamwork approaches carry assumptions that make them incompatible with real human-AI teaming scenarios. The most fundamental of these is the assumption that the *ad hoc* agent can observe, at some stage, either the full state of the environment or the real-time actions of its teammates. In human-robot or human-AI collaboration, neither is generally available: the environment is only partially observable, and a human teammate's true intentions and planned actions are internal cognitive states that cannot be directly read. A second, more technical limitation is that some approaches require the environment to be small enough to be represented as an explicit tabular model — a constraint that rules out the complex, dynamic environments where hybrid human-AI teams actually operate.

Current approaches addressing team identification in *ad hoc* teamwork can be characterised by which of these limitations they address, and which they retain. Seminal works such as PLASTIC [Barrett et al., 2017] assume both full state observability and observable teammate actions. BOPA [Ribeiro et al., 2021] relaxes the action observability requirement but still requires full state information, PO-GPL [Rahman et al., 2023] and FEAT [Fang et al., 2024] conversely handle partial state observability but retain the assumption of observable teammate actions. ODITS [Gu et al., 2021] removes both assumptions at evaluation time but requires a fully observable pre-training phase alongside each team — a requirement that is arguably incompatible with the spirit of *ad hoc* teamwork and that is rarely feasible when teammates are humans. ATPO [Ribeiro et al., 2023a] is the only approach that removes the requirement for observable teammate actions or states at any stage, but does so via tabular POMDP models hand-crafted by human experts, restricting its applicability to environments with at most a few thousand states and observations.

To address both limitations simultaneously, we propose RecBayes, a Recurrent Bayesian approach to *Ad Hoc* teamwork. RecBayes replaces the hand-crafted tabular models of prior work with a recurrent Bayesian classifier trained directly on trajectories collected during previous partial observations of each known team-task. This design choice provides two key benefits: (i) it removes the requirement for any privileged information — neither environment states nor teammate actions are ever needed, at any stage — and (ii) it scales naturally to domains with arbitrarily large state and observation spaces, since the classifier operates on learned latent representations rather than explicit tabular entries. To the best of our knowledge, RecBayes is the first approach to *ad hoc* teamwork that satisfies both properties simultaneously.

We evaluate RecBayes in two benchmark multi-agent domains from the *ad hoc* teamwork literature — Level-Based Foraging and Predator-Prey — scaled to environments containing up to $1M$ states and $2^{125}$ possible observations, under varied team and task configurations. Our results show that RecBayes is both accurate at identifying the correct team and task from partial observations alone, and capable of assisting those teams in completing their objectives at near-oracle performance levels. We discuss the implications of these results for the design of *ad hoc* agents deployable in real hybrid

human-AI settings, and we outline the open challenges that remain between this work and full deployment with human teammates.

## 2. Related Work

*Ad hoc teamwork.* *Ad hoc* teamwork literature can be framed according to how it addresses three key sub-problems: (i) task identification, (ii) team identification and (iii) planning/execution [Melo and Sardinha, 2016]. Seminal works [Stone and Kraus, 2010, Barrett and Stone, 2011] started by assuming that both the team and the task were known beforehand, with planning and execution as their only focus, and team identification was later approached by Chakraborty and Stone [2013]. Following Melo and Sardinha [2016]'s formalisation of *ad hoc* teamwork, Barrett et al. [2017] introduced the PLASTIC framework, allowing *ad hoc* agents to identify known teams by observing their actions in fully observable environments. This was followed by AATEAM [Chen et al., 2020], which uses an attention mechanism to generalise to unknown teams, and TEAMSTER [Ribeiro et al., 2023b], which continuously updates a shared world model to efficiently adapt to new teammates. All three require both observable teammate actions and full state observability.

Relaxing these requirements individually, Ribeiro et al. [2021] propose BOPA, which handles unobservable teammate actions but requires full state observability, while Rahman et al. [2023] propose PO-GPL, which handles partial state observability but requires observable teammate actions. Fang et al. [2024] extend PO-GPL with FEAT, using meta-learning to remove the need for *a priori* team sets while retaining the assumption of observable teammate actions. Gu et al. [2021] propose ODITS, which handles partial observations at evaluation time but requires a fully observable pre-training phase. Ribeiro et al. [2023a] propose ATPO, which removes all requirements for privileged information but relies on tabular POMDP models hand-crafted by domain experts, limiting applicability to environments with up to $4,831$ states and $1,731$ observations. In follow-up work, Ribeiro et al. [2024] demonstrate these methods in real-world partially observable domains involving human-robot interaction, motivating the need for approaches that scale beyond the tabular regime.

RecBayes builds directly on the line started by ATPO, removing the tabular bottleneck while preserving the core property that no privileged information is ever required.

*Hybrid human-AI teaming.* The challenge of deploying AI agents into teams that include humans has attracted significant attention across robotics, human-computer interaction and AI [Akata et al., 2020, Dafoe et al., 2020]. A recurring theme in this literature is the difficulty of modelling human teammates: human behaviour is highly variable, intentions are not directly observable, and humans cannot be expected to pre-coordinate with an AI system [Nikolaidis et al., 2017, Mirsky et al., 2022]. These properties make *ad hoc* teamwork a natural computational framework for hybrid human-AI teams, a connection noted by several authors [Ribeiro et al., 2024, Mirsky et al., 2022] but not yet fully exploited in practice. Our work contributes to closing this gap by providing an approach that lifts the observability constraints that have so far prevented *ad hoc* teamwork methods from being applicable in realistic hybrid settings.

## 3. Preliminaries

We frame *ad hoc* teamwork according to Melo and Sardinha [2016] and Ribeiro et al. [2023a], dealing not only with team identification, but also task identification. We consider partially observable domains where *ad hoc* agents are placed in an environment $\varepsilon$ containing an underlying team performing an underlying task. The *ad hoc* agent is given the goal of identifying the most-likely team-task combination $k$ from a set of $K$ possible team-tasks, and acting accordingly to efficiently assist the team in solving the task. Ribeiro et al. [2023a] achieve this by explicitly modelling each team-task $k \in \{1,\dots,K\}$ and respective environment $\varepsilon^k$ as a POMDP $m^k = (\mathscr{X}^k, \mathscr{A}, \mathscr{Z}, P^k, O^k, r^k)$, where $\mathscr{X}^k$ represents the state space, $\mathscr{A}$ the *ad hoc* agent's action space, required equal for all team-tasks, $\mathscr{Z}$ the *ad hoc* agent's observation space, required equal for all team-tasks, $P^k$ the transition probabilities modelling both the world's dynamics as well as the behaviour of the team, $O^k$ the observation probabilities, $r^k$ the reward function.

Having set up a set of team-task models $\mathscr{L} = \{m^1,\dots,m^K\}$ containing all $K$ POMDPs, Ribeiro et al. [2023a] set up a respective best-response policy library $\Pi = \{\pi^1,\dots,\pi^K\}$, a common practice in *ad hoc* teamwork, which the authors compute relying on the PERSEUS algorithm [Spaan and Vlassis, 2005].

Then, when deployed at evaluation time to an unknown environment $\varepsilon$, without knowing its underlying team and task $k$, the *ad hoc* agent starts by defining a prior distribution $p_0(k)$ for each model, which it then updates at each time step using the observation $z_{t+1}$ and the *ad hoc* agent's own action $a_t$ as evidence, i.e.

$$p_{t+1}(k) = p_t(k) \cdot \frac{1}{\rho} \cdot \sum_{x,y \in \mathscr{X}^k} O^k(z_{t+1} \mid y, a_t) \cdot P^k(y \mid x, a_t) \cdot b_t^k(x) \cdot \pi_t(a_t \mid h_t) \tag{1}$$

where $\rho$ is a normalisation constant, $b_t^k(x)$ the belief that the agent is in state $x$ at time step $t$ if the POMDP is $m^k$, and $\pi_t(a_t \mid h_t)$ the probability of the agent choosing action $a_t$ given the history $h_t = \{a_0, z_1, r_1, \dots, a_{t-1}, z_t, r_t\}$.

Finally, the authors use the probability of each model, $p_t(k)$, as a weight to compute the agent's final policy, i.e.

$$\pi_t(a \mid h_t) = \sum_{k=0}^{K} \pi^k(a \mid h_t) \cdot p_t(k) \tag{2}$$

## 4. Recurrent Bayesian Ad Hoc Teamwork

We now describe our proposed recurrent Bayesian algorithm for *ad hoc* teamwork. As shown by Equation 1, identifying the most-likely team-task is possible via Bayesian updates if the agent has access to the transition and observation probabilities of the environment. Since these tabular models are unfeasible to set up for environments with large state and observation spaces, our hypothesis is that trajectories collected during previous interactions with individual team-tasks $k$ may contain enough information to approximate the Bayesian update from Eq. 1 by framing it as a recurrent classification problem.

Specifically, when interacting with previous team-tasks $k \in \{1,\dots,K\}$, we collect $T$ trajectories $\text{traj}^k = \{\tau_1^k,\dots,\tau_T^k\}$, one set for each team-task, with $\tau_i^k = (a_0, z_1, a_1, \dots, a_{L-1}, z_L)$,

**Algorithm 1** Trajectory Collection for Previous Team-Task Combinations

**Input:**
Environments where each team-task is present $\{\varepsilon^1, \ldots, \varepsilon^K\}$
Number of trajectories to collect per environment $T$
Max trajectory length $L$
**Output:**
Trajectory buffers $\{traj^1, traj^2, \ldots, traj^K\}$
Trained policies $\{\pi_{\psi^1}, \ldots, \pi_{\psi^K}\}$
**for** $k = 1$ to $K$ **do**
  Initialize policy $\pi_{\psi^k}$ and trajectory buffer $traj^k$
  **for** $t = 1$ to $T$ **do**
    Reset $\varepsilon^k$ and trajectory $\tau = \emptyset$
    **for** $l = 1$ to $L$ **do**
      Observe $\phi(z_t)$
      Execute action $a_l \sim \pi_{\psi^k}(\phi(z_t), h_t)$
      Observe $\phi(z_{l+1})$ and reward $r_{l+1}$
      Append $(a_l, z_{l+1}, r_{l+1})$ to $\tau$
    **end for**
    Append $\tau$ to $traj^k$
    Update policy $\pi_{\psi^k}$ using trajectories in $traj^k$
  **end for**
**end for**
**return** $\{traj^1, \ldots, traj^K\}$, $\{\pi_{\psi^1}, \ldots, \pi_{\psi^K}\}$

$a_{t-1}$ denoting the actions executed by the *ad hoc* agent and $z_t$ its partial observations. After collecting the trajectories, we train a parametrised recurrent Bayesian classifier $\hat{p}_\theta$ using the labelled dataset $\mathscr{D} = \{(\tau_1, k_1), \ldots, (\tau_N, k_N)\}$, where each datapoint $(\tau_i, k_i)$ represents a trajectory and its associated team-task. After training, $\hat{p}_\theta$ is optimised to approximate a distribution over all teams and tasks $k \in \{1, \ldots, K\}$, when recurrently given evidences $(a_{t-1}, z_t)$ as input. Algorithm 1 details the process for collecting the trajectories when interacting with different teams performing potentially different tasks, alongside the training of the respective best-response policies for each one. Algorithm 2 then details how these trajectories can be used to train the recurrent Bayesian classifier $\hat{p}_\theta$ by formulating it as a recurrent classification problem. A natural consequence of this formulation is that we can relax the requirement made by Ribeiro et al. [2023a] that the observation's explicit tabular entry $z_t$ is required, being instead described by an arbitrary set of features $\phi(z_t)$.

Given the set of trained best-response policies $\{\pi_{\psi^1}, \ldots, \pi_{\psi^K}\}$, a common method in the current *ad hoc* teamwork literature [Barrett et al., 2017, Chen et al., 2020, Gu et al., 2021, Ribeiro et al., 2023b, Rahman et al., 2023, Ribeiro et al., 2023a], alongside the trained recurrent Bayesian classifier $\hat{p}_\theta$ able to compute a distribution over all known team-tasks, the *ad hoc* agent's final policy can be computed by adapting Equation 2 as follows:

$$\pi_t(a \mid \phi(z_t)) = \sum_{k=0}^{K} \pi_{\psi^k}(a \mid \phi(z_t), h_t^k) \cdot \hat{p}_\theta(k \mid \phi(z_t), h_t) \tag{3}$$

**Algorithm 2** Recurrent Bayesian Classifier Training

**Input:**
Trajectory buffers $\{traj^1, traj^2, \ldots, traj^K\}$, respectively labelled $k \in \{1, 2, \ldots, K\}$
Number of epochs $E$
Learning rate $\alpha$
Batch size $B$
**Output:**
Trained Recurrent Bayesian Classifier $\hat{p}_\theta$ with parameters $\theta$
Initialize classifier $\hat{p}_\theta$ with random parameters $\theta$
Create labelled dataset $\mathscr{D} = \{(\tau_1, k_1), \ldots, (\tau_N, k_N)\}$ using $\{traj^1, traj^2, \ldots, traj^K\}$
**for** epochs $= 1$ to $E$ **do**
  Sample batch $\{(\tau_1, k_1), \ldots, (\tau_B, k_B)\}$ from $\mathscr{D}$
  Compute logits $\{z_1, \ldots, z_B\} = \hat{p}_\theta(\{\tau_1, \ldots, \tau_B\})$
  Compute probabilities $\{p_1, \ldots, p_B\} = \text{softmax}(\{z_1, \ldots, z_B\})$
  Compute categorical cross-entropy loss $\mathscr{L}(\theta) = -\frac{1}{B}\sum_{b=1}^{B}\sum_{k=1}^{K} p_b^k \log(p_b^k)$
  Compute gradient $\nabla_\theta \mathscr{L}(\theta)$
  Update parameters $\theta \leftarrow \theta - \alpha \nabla_\theta \mathscr{L}(\theta)$
**end for**
**return** Trained Recurrent Bayesian Classifier $\hat{p}_\theta$

**Algorithm 3** RecBayes: Recurrent Bayesian *Ad Hoc* Teamwork

**Input:**
Unknown environment $\varepsilon$
Trained Bayesian classifier $\hat{p}_\theta$
Set of known policies $\{\pi_{\psi^1}, \ldots, \pi_{\psi^K}\}$ for each team-task
Number of evaluation episodes $E$
Initialize prior distribution $p_0$
**for** *episode* $= 1$ to $E$ **do**
  Reset environment $\varepsilon^k$
  **while** not terminal **do**
    Observe $\phi(z_t)$
    Compute policy $\pi_t(a \mid \phi(z_t)) = \sum_{k=0}^{K} \pi_{\psi^k}(a \mid \phi(z_t), h_t^k) \cdot p_t$
    Execute action $a_t \sim \pi_t$
    Observe $\phi(z_{t+1})$ and reward $r_{t+1}$
    Update priors $p_{t+1} = \hat{p}_\theta(a_t, z_{t+1}; h_t)$
  **end while**
**end for**

where $h_t^k$ represents the latent recurrent state kept by policy $\pi_{\psi^k}$ and $h_t$ the latent recurrent state kept by the Bayesian classifier $\hat{p}_\theta$. Putting it all together, Algorithm 3 summarises the full algorithm for performing *ad hoc* teamwork, including action selection at evaluation time.

## 5. Experimental Setup

In our evaluation, we aim to answer two main research questions:

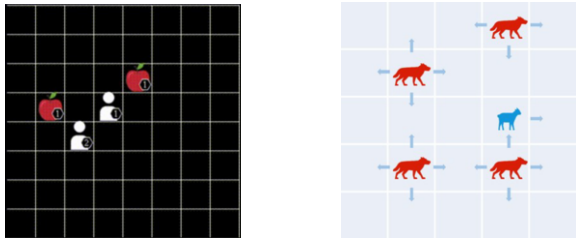
- **RQ1**: Is RecBayes efficient in assisting the teams solving the tasks?
- **RQ2**: Is RecBayes effective in identifying the teams and tasks from observations alone?

We follow the evaluation procedure for *ad hoc* teamwork proposed by Stone et al. [2010], which first evaluates a team of original teammates and afterwards replaces one of the members, at random, with the *ad hoc* agent to be evaluated. A single trial consists of running an episode where an *ad hoc* agent is deployed on-the-fly and stays with the team until the task is solved. When the episode is over, we register the total number of steps it took to complete. Each experiment — defined by one agent, deployed into one team performing one task, in one domain, for one world-size — consists of 16 trials.

We now describe our domains, different teams, different tasks and baseline agents.

### 5.1. Domain Descriptions

We evaluate our approach in two benchmark domains from the *ad hoc* teamwork literature — the Level-Based Foraging (LBF) domain and the Predator-Prey (PP) domain (Figure 1), scaled up to provide a large enough state space and adapted for partial observability. Although these domains are commonly used benchmarks within both the multi-agent systems and *ad hoc* teamwork literature, they are often instantiated in relatively simple or small variations, such as fully observable $5 \times 5$ worlds. To showcase the effectiveness of our approach in arbitrarily large state spaces, we conduct experiments in variations ranging from $7 \times 7$ worlds as the smallest, containing $117,649$ total states, up to $10 \times 10$, containing a total of $970,200$ states as the largest. We introduce partial observability by adapting the observation function from Santos et al. [2022], which fixes a limited field-of-view around the agent containing several channels for identifying other agents, walls, and other information such as levels. In all experiments and world sizes, we keep a field-of-view of size $5 \times 5$, providing a large and non-unidimensional observation space containing up to $2^{125}$ different observations. Figure 2 showcases two example observations from the Level-Based Foraging domain.



(a) LBF (b) PP

**Figure 1.** Visualisation of two example states in each of our benchmark environments, (a) Level-Based Foraging (LBF) and (b) Predator-Prey (PP).

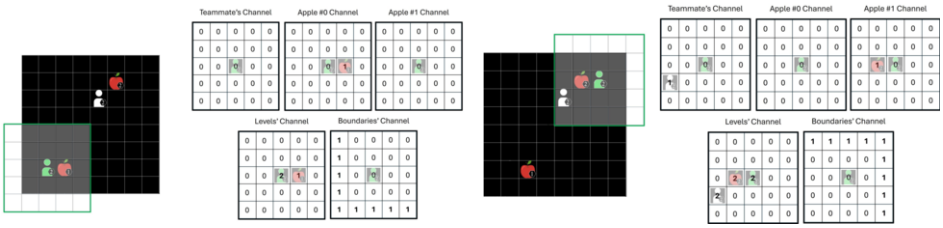



**Figure 2.** Example tuple observations of shape $5 \times 5 \times 5$. Five different channels of size $5 \times 5$ provide the agent a limited field-of-view which detects both entities and other information in the near environment. Each other agent is given a separate channel since indexes are relevant in these domains, used for instance for stalemate resolution or other decision processes.

### *5.2. Identification Sub-Problems*

In our evaluation, we conduct three distinct sets of experiments in both domains — (i) **Team Identification**, where the task is fixed and the *ad hoc* agent has to identify and assist the correct team, (ii) **Task Identification**, where the team is fixed and the *ad hoc* agent has to identify and assist in performing the correct task, and (iii) **Task & Teammate Identification**, where neither the task nor the team is fixed, requiring the *ad hoc* agent to identify and assist the team in completing the tasks. We now describe the possible teams and tasks.

*Teams* Teams from both domains can have three different strategies. The first strategy (named Greedy) finds the closest target and moves towards it without taking into account any obstacles in the field. This strategy makes no assumptions regarding its teammates. The second strategy (named Teammate Aware) picks the target with the highest value — highest level in LBF and the prey closest to all predators in PP — and moves towards it while taking into account obstacles such as teammates. This strategy assumes teammates share the same decision-making process. Finally, the third strategy (named Probabilistic Destinations) also selects the highest-value target while avoiding obstacles, but searches for the positions of teammates to compute a plan that encircles the target, preventing collisions and, in PP, the prey from escaping. Like the second strategy, this one also assumes teammates share the same decision-making.

*Tasks* In both domains, tasks are defined by the cardinal directions assigned to each individual team member when approaching a target. For example, in Level-Based Foraging, where agents must coordinate to pick up apples spread throughout the field, only a single agent may approach from the north, another from the south, another from the west, and another from the east. We also allow an additional task where no coordinates are assigned.

### *5.3. Baseline Agents*

We evaluate a total of six baseline agents in the role of the *ad hoc* agent: two ablations of our approach, (i) one relying on a set of model-free reinforcement learning policies (**RecBayes-MF**) and (ii) another relying on a set of model-based reinforcement learning policies (**RecBayes-MB**); (iii) a teammate following the same policy as the original team (**Original Teammate**); (iv) an oracle model-free RL agent which knows the cor-

**Table 1.** Average number of steps to solve an episode in all experiments, the smaller the better. For each baseline, team, task, domain, world-size, we repeat trials 16 times. Agents relying on deep learning models (Model-Free, Model-Based, RecBayes-MF and RecBayes-MB) are evaluated two additional times to account for three total parameter initialisations. In total, for all variations and baselines, 12544 trials were conducted in our experiments.

| | **Original Teammate** | | **Model-Free RL** (Knows Task/Team) | | **Model-Based RL** (Knows Task/Team) | | **RecBayes MF** | | **RecBayes MB** | | **Random Policy** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Level-Based Foraging (7x7)** | | | | | | | | | | | | |
| Task Identification | 6.87 | (±1.86) | 7.56 | (±2.42) | 8.13 | (±5.49) | 13.02 | (±10.47) | 15.35 | (±17.15) | 222.6 | (±222.73) |
| Team Identification | 8.57 | (±8.77) | 7.68 | (±5.06) | 7.31 | (±4.91) | 8.85 | (±7.5) | 10.24 | (±8.38) | 86.37 | (±82.44) |
| Task & Team Identification | 7.69 | (±6.3) | 7.62 | (±3.92) | 7.73 | (±5.24) | 11.96 | (±9.12) | 14.06 | (±11.43) | 168.11 | (±192.19) |
| **Predator-Prey (7x7)** | | | | | | | | | | | | |
| Task Identification | 9.55 | (±4.05) | 11.22 | (±6.1) | 9.79 | (±3.81) | 21.36 | (±15.14) | 17.34 | (±9.91) | 270.66 | (±280.81) |
| Team Identification | 16.69 | (±18.6) | 10.19 | (±7.16) | 10.11 | (±6.55) | 10.71 | (±7.47) | 10.64 | (±6.83) | 120.48 | (±123.13) |
| Task & Team Identification | 12.61 | (±13.04) | 10.71 | (±6.67) | 9.95 | (±5.35) | 16.75 | (±11.62) | 15.67 | (±9.28) | 209.35 | (±241.46) |
| **Level-Based Foraging (10x10)** | | | | | | | | | | | | |
| Task Identification | 9.34 | (±2.98) | 33.67 | (±50.32) | 21.52 | (±25.42) | 66.53 | (±90.49) | 33.86 | (±38.88) | 411.84 | (±266.54) |
| Team Identification | 11.48 | (±7.37) | 15.65 | (±21.42) | 15.15 | (±10.68) | 19.27 | (±18.59) | 25.88 | (±19.86) | 230.22 | (±187.08) |
| Task & Team Identification | 10.27 | (±5.44) | 26.69 | (±42.5) | 18.79 | (±20.69) | 37.43 | (±40.61) | 35.47 | (±41.48) | 338.96 | (±253.98) |
| **Predator-Prey (10x10)** | | | | | | | | | | | | |
| Task Identification | 12.11 | (±4.91) | 17.52 | (±10.25) | 16.27 | (±8.2) | 78.65 | (±185.69) | 59.38 | (±85.73) | 457.65 | (±293.54) |
| Team Identification | 42.23 | (±73.61) | 31.75 | (±29.52) | 20.17 | (±15.83) | 32.71 | (±46.42) | 18.77 | (±12.42) | 271.11 | (±242.77) |
| Task & Team Identification | 25.02 | (±50.58) | 25.0 | (±23.64) | 17.94 | (±12.23) | 26.29 | (±19.69) | 34.41 | (±24.68) | 353.03 | (±281.9) |

rect team-task beforehand (**Model-Free RL**); (v) an oracle model-based RL agent which knows the correct team-task beforehand (**Model-Based RL**); and (vi) a **Random Policy**.

We evaluate both the model-free and model-based approaches to demonstrate that RecBayes is viable with either paradigm, which alternate in their domination as the state-of-the-art across different domains. Although our approach is not coupled with any particular algorithm, we chose PPO [Schulman et al., 2017] as our model-free algorithm and DreamerV3 [Hafner et al., 2023] as our model-based one. Hyperparameters for both were fine-tuned using Optuna [Akiba et al., 2019] and are fully available in file `hyperparameters.yaml` in our repository at **Anonymised for Review**.

### *5.4. Model Implementations*

Both our Bayesian Classifier $\hat{p}_\theta$ and policies $\pi_\psi$ require an input module capable of handling partial tuple-based observations of shape $5 \times 5 \times 5$. To extract strong latent representations usable by both the classifier and the best-response policies, a natural design choice is to implement these input modules as recurrent convolutional encoders, which we follow in this work. Figure 4 in the Appendix provides an overview of the architecture, manually fine-tuned. All hyperparameters are fully available in file `hyperparameters.yaml` in our repository.

## 6. Results

We conduct our experiments with all six agents, on all three sets, on both domains, on both space sizes ($|\mathscr{X}| = 117649, |\mathscr{Z}| = 2^{125}$ and $|\mathscr{X}| = 970200, |\mathscr{Z}| = 2^{125}$). Agents relying on deep learning models are evaluated three total times with different parameter initialisations. In total, for all variations and baselines, 12544 trials were conducted in our experiments. We now break down our results by research question.

**Table 2.** Results from Table 1 normalised between the ones obtained by the Original Teammate and the Random Policy, the higher the better. Values equal and above 1.0 (highlighted) mean an agent has matched or outperformed the Original Teammate, while values below 0.0 mean an agent performed worse than the random policy.

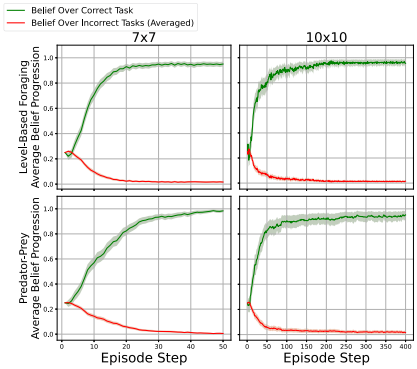
| | **Original Teammate** | **Model-Free RL** (Knows Task/Team) | **Model-Based RL** (Knows Task/Team) | **RecBayes MF** | **RecBayes MB** | **Random Policy** |
|---|---|---|---|---|---|---|
| **Level-Based Foraging (7x7)** | | | | | | |
| Task Identification | 1.00 | 0.99 | 0.99 | 0.97 | 0.96 | 0.00 |
| Team Identification | 1.00 | **1.01** | **1.02** | 0.99 | 0.98 | 0.00 |
| Task & Team Identification | 1.00 | 1.00 | **1.00** | 0.97 | 0.96 | 0.00 |
| **Predator-Prey (7x7)** | | | | | | |
| Task Identification | 1.00 | 0.99 | **1.00** | 0.95 | 0.97 | 0.00 |
| Team Identification | 1.00 | **1.06** | **1.06** | **1.06** | **1.06** | 0.00 |
| Task & Team Identification | 1.00 | **1.02** | **1.01** | 0.98 | 0.99 | 0.00 |
| **Level-Based Foraging (10x10)** | | | | | | |
| Task Identification | 1.00 | 0.94 | 0.97 | 0.86 | 0.94 | 0.00 |
| Team Identification | 1.00 | 0.98 | 0.98 | 0.96 | 0.93 | 0.00 |
| Task & Team Identification | 1.00 | 0.95 | 0.97 | 0.92 | 0.92 | 0.00 |
| **Predator-Prey (10x10)** | | | | | | |
| Task Identification | 1.000 | 0.98 | 0.991 | 0.85 | 0.89 | 0.00 |
| Team Identification | 1.00 | **1.05** | **1.10** | **1.04** | **1.10** | 0.00 |
| Task & Team Identification | 1.00 | **1.00** | **1.02** | 0.99 | 0.97 | 0.00 |

### *6.1. Research Question #1: Is RecBayes efficient in assisting the teams solving the tasks*

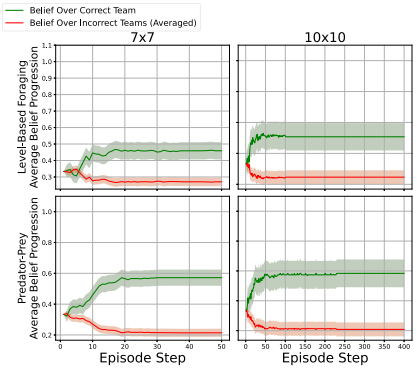
To answer our first research question, we compare the results obtained by RecBayes against the other baselines. Table 1 presents the average numbers of steps to complete an episode for all baselines on all settings. As done by Ribeiro et al. [2023a], to better analyse the impact of each agent, we normalise the results between the ones achieved Original Teammate and the Random Policy. This way, if an agent has a relative performance above 1.0, it means it has outperformed the Original Teammate, while relative performances below 0.0 mean the agent has underperformed the Random Policy. We consider a relative performance above 0.90 to be near optimal. Table 2 presents the normalised results.

The first observation we can make is that, as expected, the Original Teammates was never significantly outperformed. In fact, we can observe that it was matched by both RL baselines in all cases, which validates their proper set up for each specific team-task. We can also observe that there are no statistically significant differences between the Model-Free and Model-Based RL agents in all 12 cases.

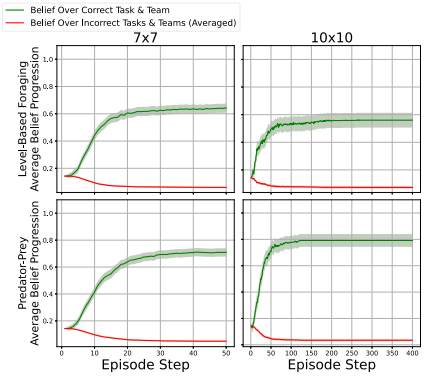
We then analyse the performances of RecBayes's ablations. RecBayes-MF was able to achieve near-optimal performance in 10 out of the 12 cases and RecBayes-MB in 11 out of the 12 cases. All three non-optimal cases were related with Task Identification, suggesting one of two hypothesis — (a) identifying the correct task is harder than identifying the correct team or (b) not having full certainty on the correct task has a higher impact on performance than not having full certainty on the correct team — which we address in the next sub-section. We note that in the larger $10 \times 10$ domains the gap between RecBayes and the oracle baselines widens modestly, a natural consequence of the increased difficulty of the classification problem at scale. Nonetheless, RecBayes consistently and substantially outperforms the random policy in all cases, confirming its practical utility. To conclude, we can positively say from these results that RecBayes is efficient in assisting the teams in solving the tasks.




(a) Task Identification




(b) Team Identification




(c) Task & Team Identification

**Figure 3.** Identification of the correct team and task by RecBayes's Bayesian Classifier $\hat{p}$. In each timestep, $\hat{p}$ outputs a probability distribution over all team-tasks, with each entry representing the belief over its respective team-task. The belief for the correct team-task is shown in green, while the average sum of the incorrect ones is shown in red.

### 6.2. *Research Question #2: Is RecBayes effective in identifying the correct team-task from observations alone?*

Figure 3 shows RecBayes's output probabilities $p_t(k)$ over the course of interactions for all three identification sub-problems, where the identified team-task is the most-likely one, i.e. $k = \arg\max_{k \in \{1,\dots,K\}} p_t(k)$. Our framing of team-task identification as a recurrent sequence classification problem is effective at identifying the correct team-task on-the-fly. Tasks are identified more quickly and with higher confidence than teams, explaining the slightly lower performance on the three non-optimal Task Identification cases in RQ1: early classifier uncertainty about the task translates more directly into suboptimal action selection than equivalent uncertainty about the team. Nonetheless, all combinations were correctly classified by episode end, demonstrating the effectiveness of our Bayesian classifier formulation.

## 7. Towards Human-AI Ad Hoc Teamwork

The experiments presented in this paper use simulated agent-controlled teammates rather than human participants, and we believe it is important to be explicit about what this means for the applicability of RecBayes in genuine hybrid human-AI settings.

Although the experiments presented in this paper use simulated agents rather than human participants, the core technical contribution of RecBayes — removing the requirement for observable teammate actions or full environment states at any stage — is precisely the property that prior *ad hoc* teamwork approaches lacked for deployment alongside human teammates. Humans are the archetypal "unknown teammate": their internal states are not directly observable, their action intentions cannot be read from the environment, and they cannot be expected to pre-coordinate with an AI system or to behave according to a fixed, known policy. The assumptions that RecBayes relaxes are therefore not arbitrary technical choices but the specific bottlenecks that have prevented *ad hoc* teamwork from being applied in hybrid settings.

That said, deploying RecBayes in a human-AI team introduces challenges that go beyond the scope of the current work. First, human teammates are not drawn from a finite known set of behavioural types in the way that the teams in our evaluation are. RecBayes, like most of the *ad hoc* teamwork literature, assumes a closed-world of known team-task combinations encountered during training. Extending it to handle the open-ended variability of human behaviour would require integration with adaptive or zero-shot coordination techniques [Mirsky et al., 2022]. Second, the trajectories used to train the Bayesian classifier in RecBayes are collected via interaction with simulated agents. In a human-AI setting, collecting comparable trajectory data would require working with actual human participants, raising practical and ethical questions about data collection, privacy, and consent. Third, evaluating *ad hoc* performance with human teammates requires metrics beyond steps-to-completion, including perceived fluency, trust, and subjective teammate quality [Nikolaidis et al., 2017].

Despite these open challenges, we believe RecBayes represents a meaningful step toward the goal of AI agents that can join and assist human teams without preparation or pre-coordination. The properties demonstrated here — scalability, identification from partial observations alone, and near-oracle collaborative performance — are necessary, if not yet sufficient, for effective hybrid human-AI teaming, with human-subject validation as the most pressing direction for future work.

## 8. Conclusion

We present RecBayes, a recurrent Bayesian approach to *ad hoc* teamwork that is agnostic to problem size and, unlike all prior approaches, never requires access to environment states or teammate actions at any stage — the precise constraints that have prevented existing methods from being applied in genuine hybrid human-AI settings. By relying solely on partially observable trajectories, RecBayes trains a recurrent Bayesian classifier to identify known teams and tasks on-the-fly, achieving near-oracle collaborative performance across domains with up to nearly $1M$ states and $2^{125}$ observations. We believe RecBayes establishes a principled and scalable foundation for *ad hoc* agents in real human-AI teams, with human-subject validation as the most pressing direction for future work.

## References

Zeynep Akata, Dan Balliet, Maarten De Rijke, Frank Dignum, Virginia Dignum, Guszti Eiben, Antske Fokkens, Davide Grossi, Koen Hindriks, Holger Hoos, et al. A research agenda for hybrid intelligence: augmenting human intellect with collaborative, adaptive, responsible, and explainable artificial intelligence. *Computer*, 53(8):18–28, 2020.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631, 2019.

S. Barrett and P. Stone. Ad hoc teamwork modeled with multi-armed bandits: An extension to discounted infinite rewards. In *Proc. AAMAS Workshop on Adaptive Learning Agents*, 2011.

S. Barrett, A. Rosenfeld, S. Kraus, and P. Stone. Making friends on the fly: Cooperating with new teammates. *Artificial Intelligence*, 242:132–171, 2017.

D. Chakraborty and P. Stone. Cooperating with a Markovian ad hoc teammate. In *Proc. 12th Int. Conf. Autonomous Agents and Multiagent Systems*, 2013.

Shuo Chen, Ewa Andrejczuk, Zhiguang Cao, and Jie Zhang. Aateam: Achieving the ad hoc teamwork by employing the attention mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7095–7102, 2020.

Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R McKee, Joel Z Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative ai. *arXiv preprint arXiv:2012.08630*, 2020.

Qi Fang, Junjie Zeng, Haotian Xu, Yue Hu, and Quanjun Yin. Learning ad hoc cooperation policies from limited priors via meta-reinforcement learning. *Applied Sciences*, 14(8):3209, 2024.

Pengjie Gu, Mengchen Zhao, Jianye Hao, and Bo An. Online ad hoc teamwork under partial observability. In *International conference on learning representations*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy P. Lillicrap. Mastering diverse domains through world models. *CoRR*, abs/2301.04104, 2023. .

F. Melo and A. Sardinha. Ad hoc teamwork by learning teammates' task. *Autonomous Agents and Multi-Agent Systems*, 30(2):175–219, 2016.

Reuth Mirsky, Ignacio Carlucho, Arrasy Rahman, Elliot Fosong, William Macke, Mohan Sridharan, Peter Stone, and Stefano V Albrecht. A survey of ad hoc teamwork: Definitions, methods, and open problems. *arXiv preprint arXiv:2202.10450*, 2022.

Stefanos Nikolaidis, David Hsu, and Siddhartha Srinivasa. Human-robot mutual adaptation in collaborative tasks: Models and experiments. *The International Journal of Robotics Research*, 36(5-7):618–634, 2017.

Arrasy Rahman, Ignacio Carlucho, Niklas Höpner, and Stefano V Albrecht. A general learning framework for open ad hoc teamwork using graph-based policy learning. *Journal of Machine Learning Research*, 24(298):1–74, 2023.

João G Ribeiro, Miguel Faria, Alberto Sardinha, and Francisco S Melo. Helping people on the fly: Ad hoc teamwork for human-robot teams. In *Progress in Artificial Intelligence: 20th EPIA Conference on Artificial Intelligence, EPIA 2021, Virtual Event, September 7–9, 2021, Proceedings 20*, pages 635–647. Springer, 2021.

João G Ribeiro, Cassandro Martinho, Alberto Sardinha, and Francisco S Melo. Making friends in the dark: Ad hoc teamwork under partial observability. In *ECAI 2023*, pages 1954–1961. IOS Press, 2023a.

João G Ribeiro, Gonçalo Rodrigues, Alberto Sardinha, and Francisco S Melo. Teamster: Model-based reinforcement learning for ad hoc teamwork. *Artificial Intelligence*, 324: 104013, 2023b.

João G Ribeiro, Luis Müller Henriques, Sérgio Colcher, Julio Cesar Duarte, Francisco S Melo, Ruy Luiz Milidiú, and Alberto Sardinha. Hotspot: An ad hoc teamwork platform for mixed human-robot teams. *Plos one*, 19(6):e0305705, 2024.

Pedro P Santos, Diogo S Carvalho, Miguel Vasco, Alberto Sardinha, Pedro A Santos, Ana Paiva, and Francisco S Melo. Centralized training with hybrid execution in multi-agent reinforcement learning. *arXiv preprint arXiv:2210.06274*, 2022.

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

Matthijs TJ Spaan and Nikos Vlassis. Perseus: Randomized point-based value iteration for pomdps. *Journal of artificial intelligence research*, 24:195–220, 2005.

Peter Stone and Sarit Kraus. To teach or not to teach?: Decision making under uncertainty in ad hoc teams. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, pages 117–124, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-0-9826571-1-9. URL http://dl.acm.org/citation.cfm?id=1838206.1838223.

Peter Stone, Gal A Kaminka, Sarit Kraus, and Jeffrey S Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

## Appendix

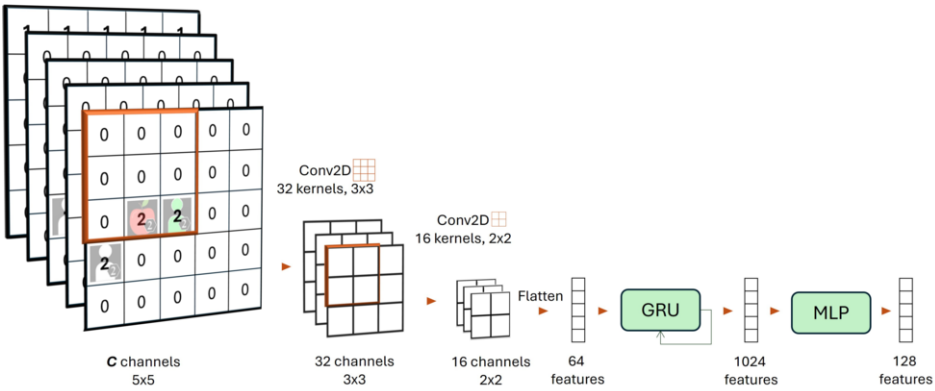



**Figure 4.** Encoder (or input) networks used both by the Bayesian classifier $\hat{p}_\theta$ and policies $\pi_\psi$ for automatic feature extraction. Observations $\phi(z_t) \in \mathbb{R}^{(5\times5\times5)}$ undergo 2D Convolutional layers, extracting a latent observation $\hat{z}_t \in \mathbb{R}^{64}$, followed by a Gated Recurrent Unit and an additional feedforward network (MLP) with two hidden layers, extracting a final latent state $\hat{x}_t \in \mathbb{R}^{128}$.