Collaboration in Ad Hoc Teamwork: Ambiguous Tasks, Roles, and Communication

Jonathan Grizou, Samuel Barrett
Peter Stone, Manuel Lopes

1. Ad Hoc Teamwork
2. Unknown Task, Team Role, Communication
3. Pursuit Domain
4. Method
5. Results
6. Limitations and future work

Ad Hoc Teamwork

Ad Hoc Teamwork

Ad Hoc Teamwork

Ad Hoc Teamwork

Ad Hoc Teamwork

- Single agent control
- Unknown teammates
- Shared goals
- No pre-coordination

Examples in humans:

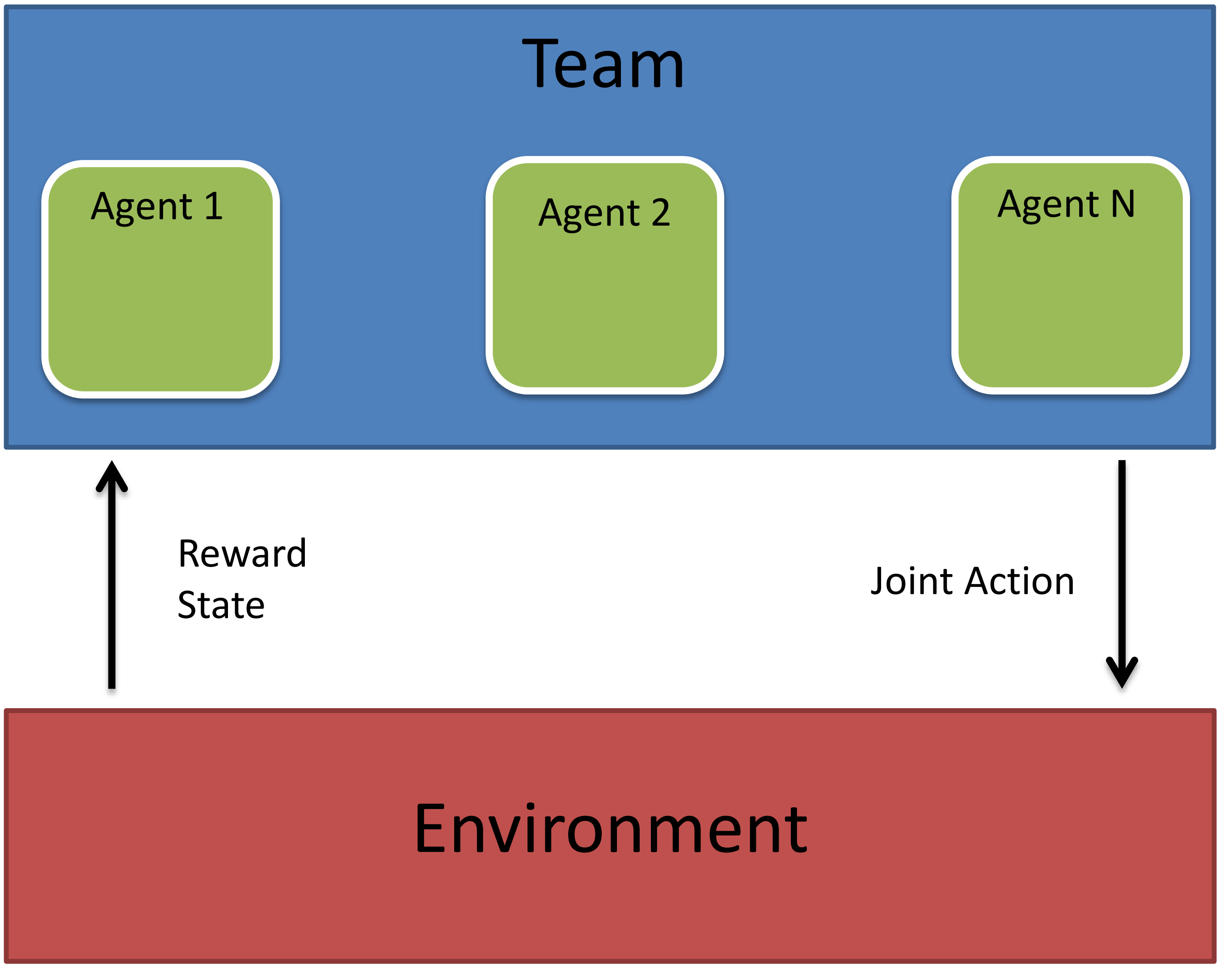
- Pick up soccer
- Accident response

Ad Hoc Teamwork

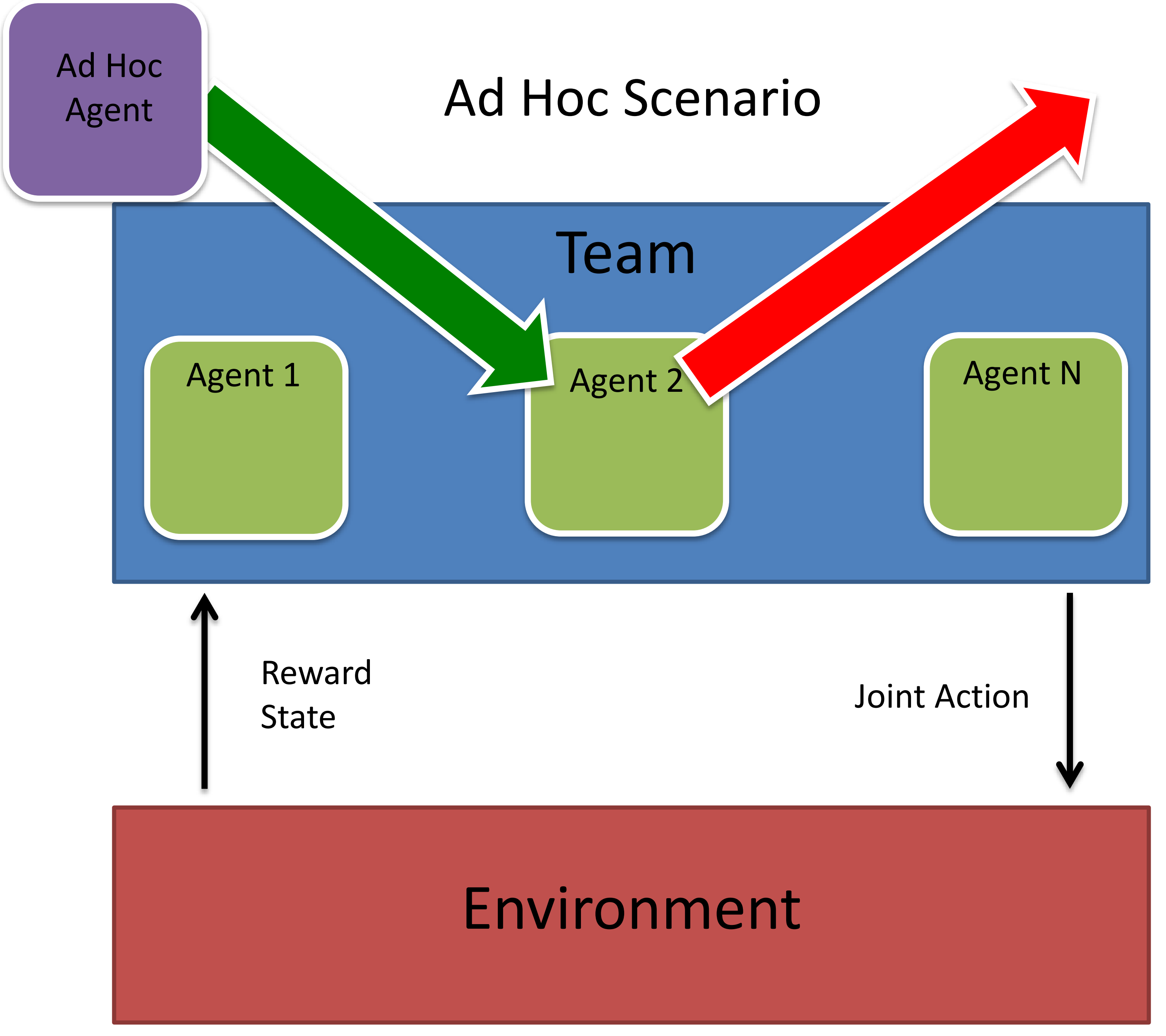
Motivations

- Agents more common and lasting longer
  (both software and hardware)
- Pre-coordination may not be possible
- Agents should be robust to various teammates
- Need to adapt quickly!

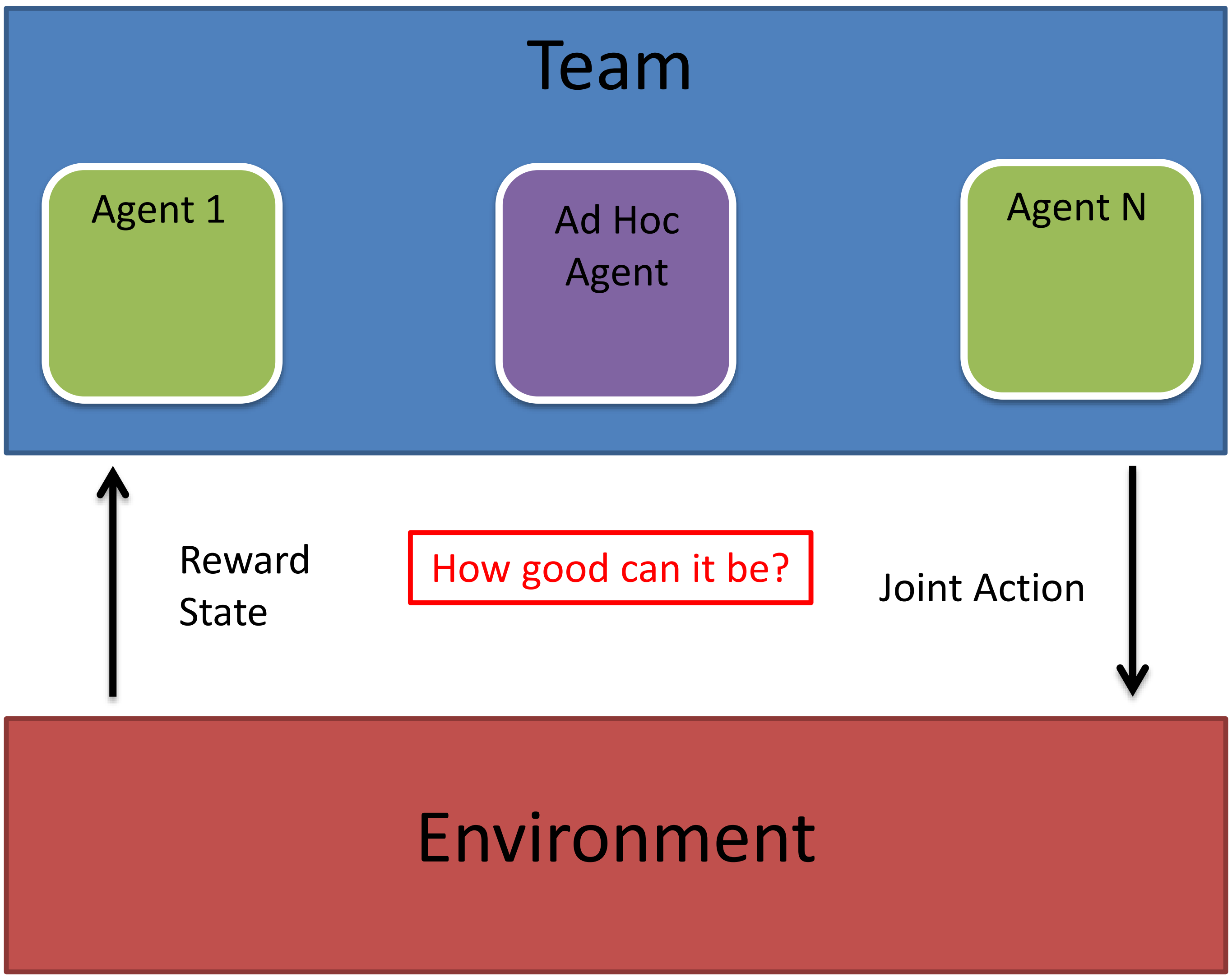
Ad Hoc Scenario



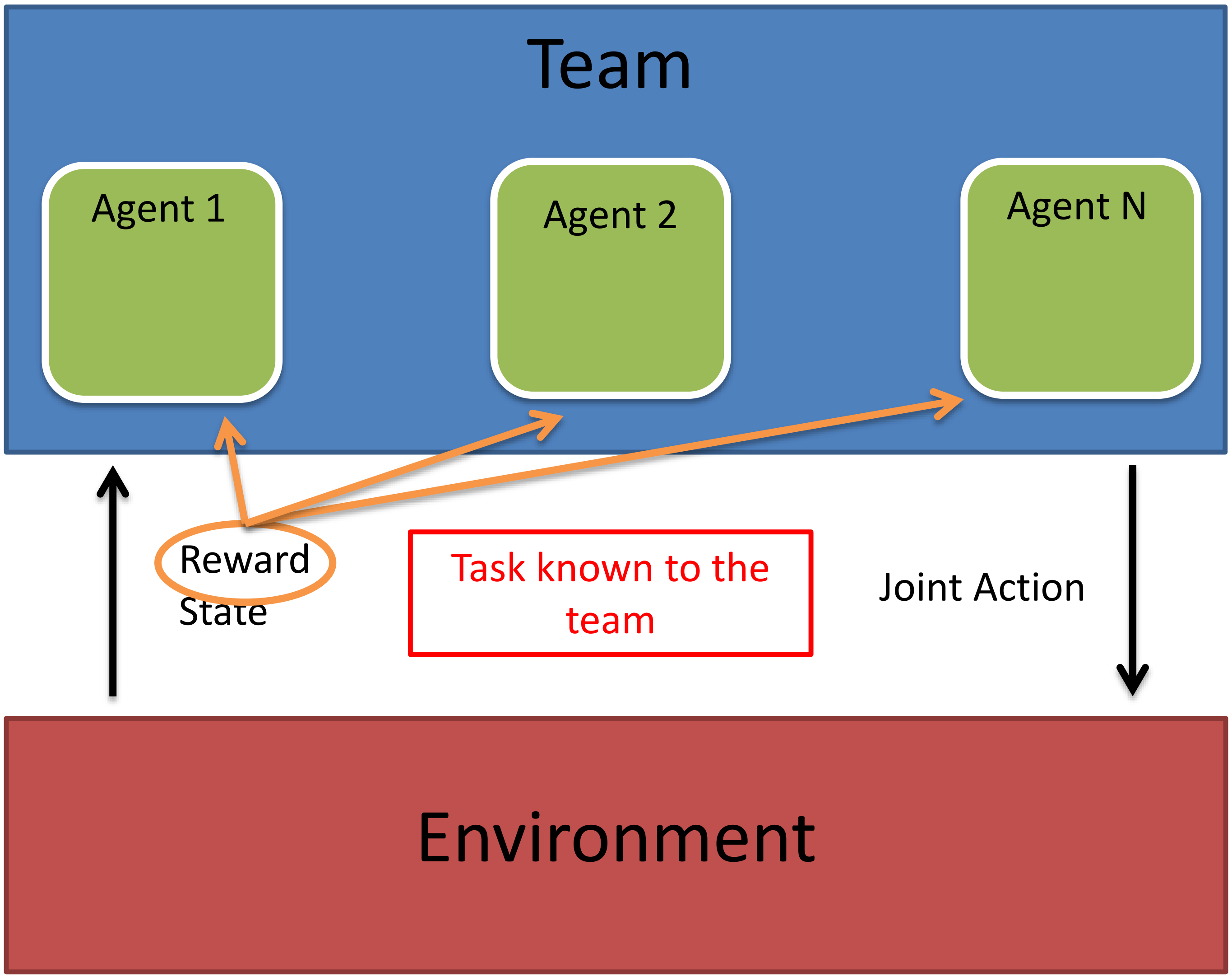
Ad Hoc Scenario



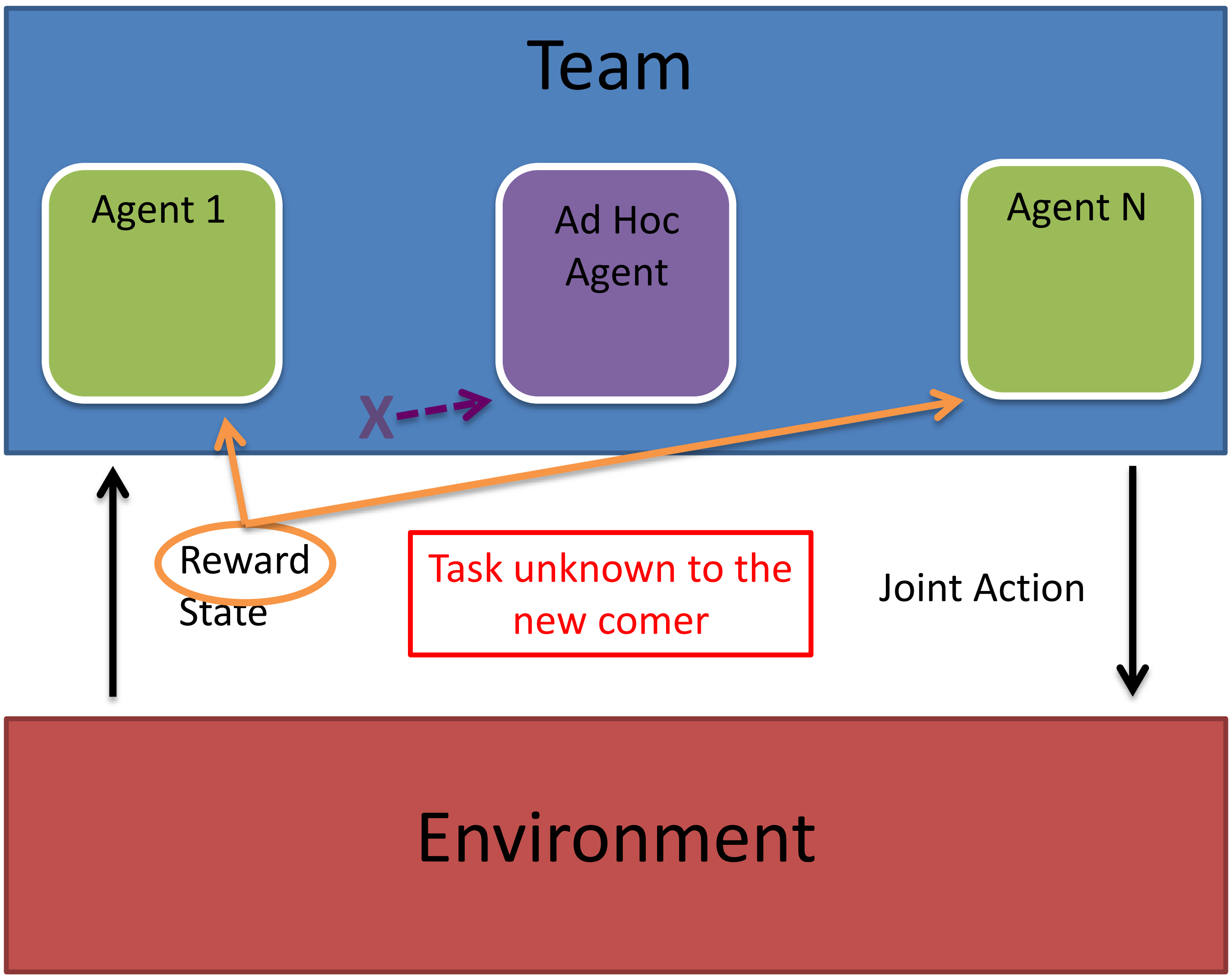
Ad Hoc Scenario



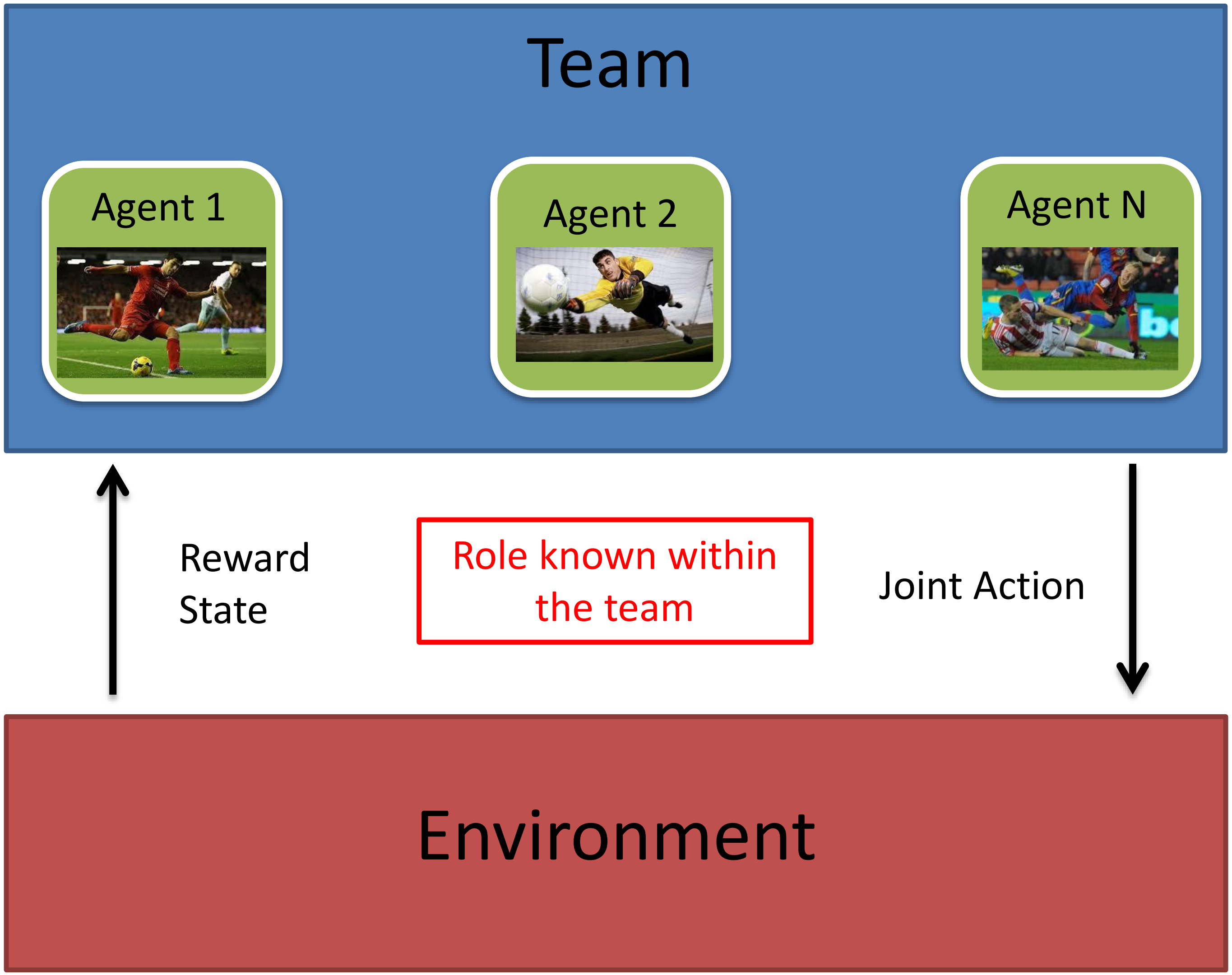
Ad Hoc Scenario



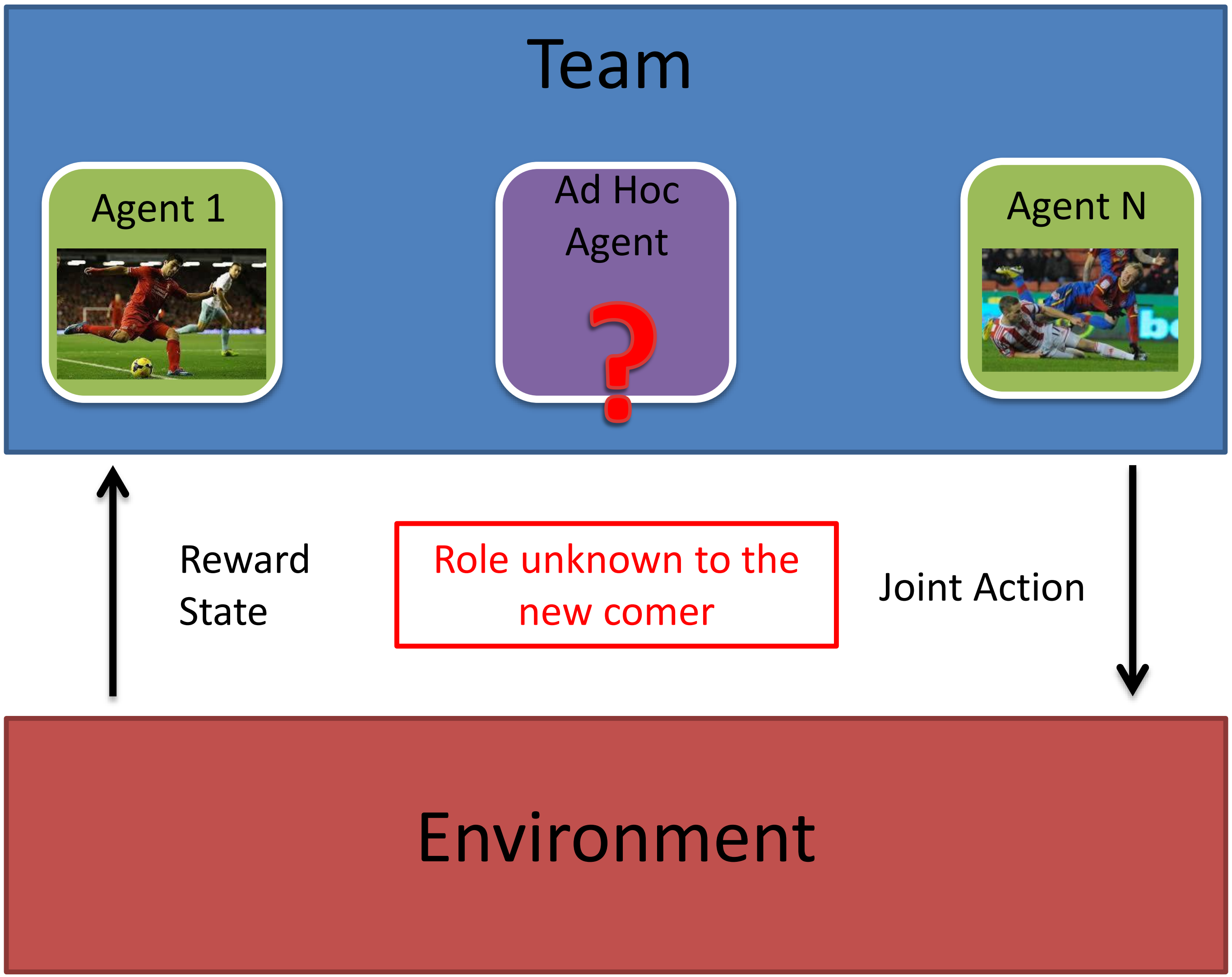
Ad Hoc Scenario



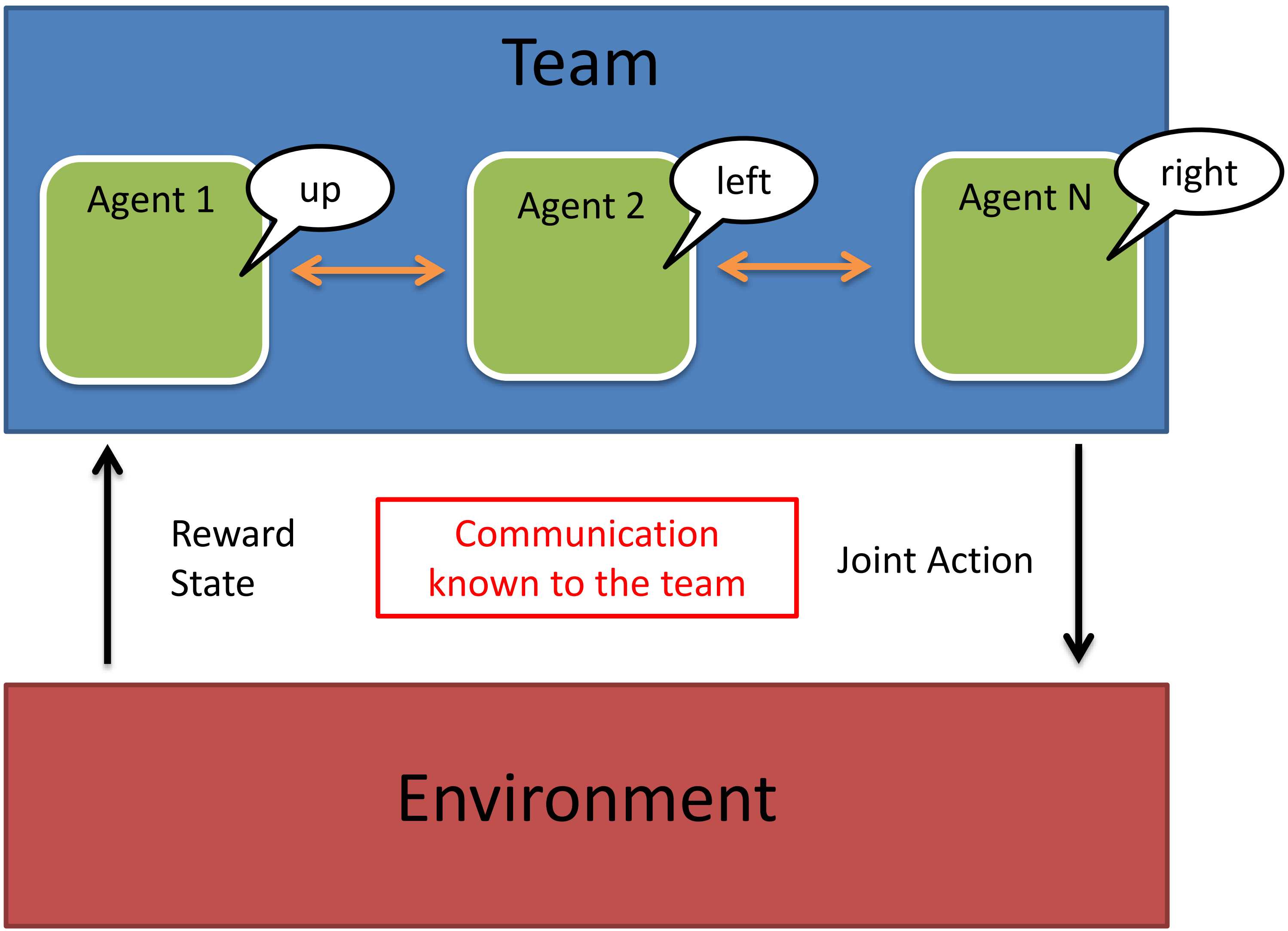
Ad Hoc Scenario



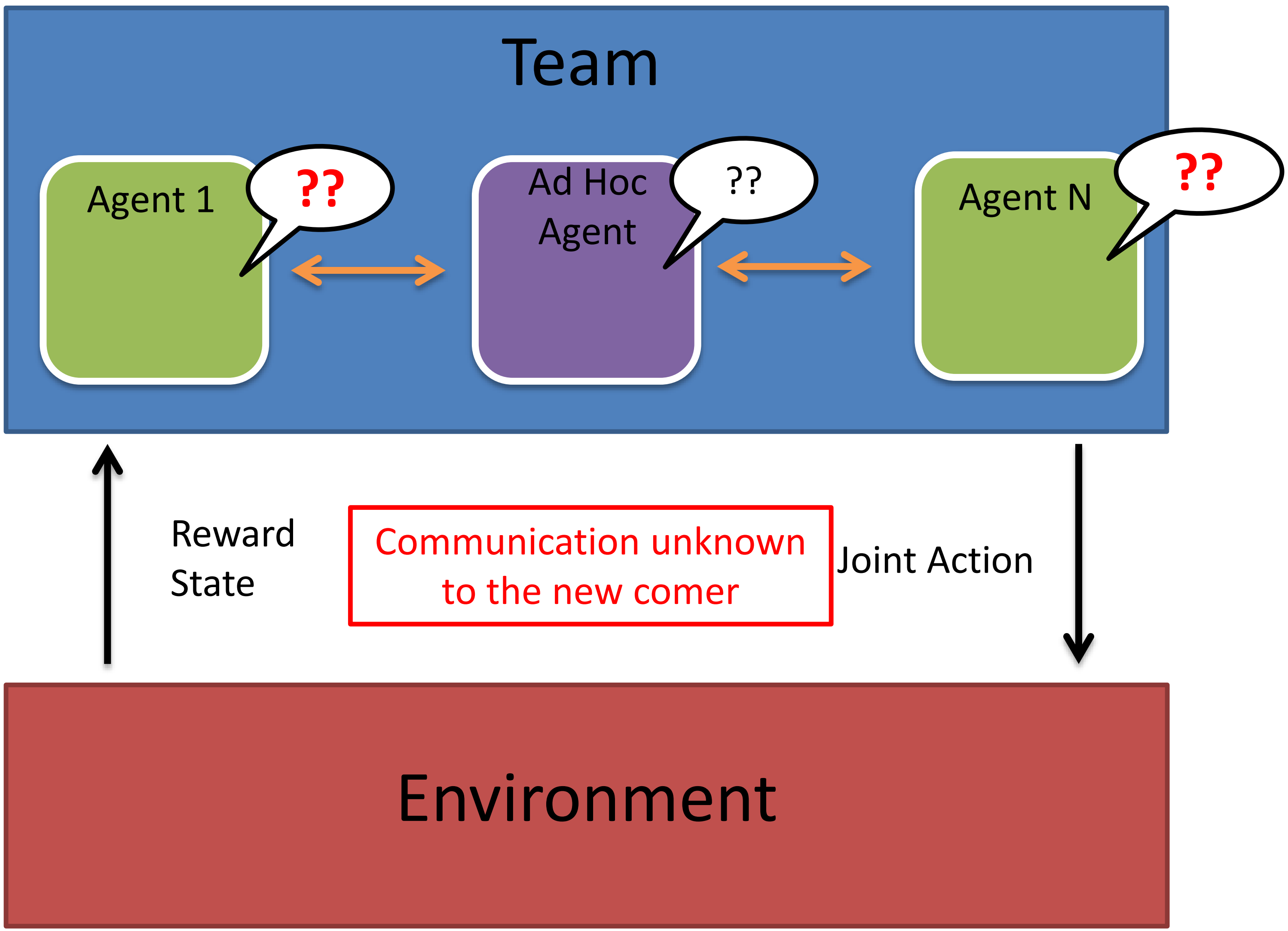
Ad Hoc Scenario



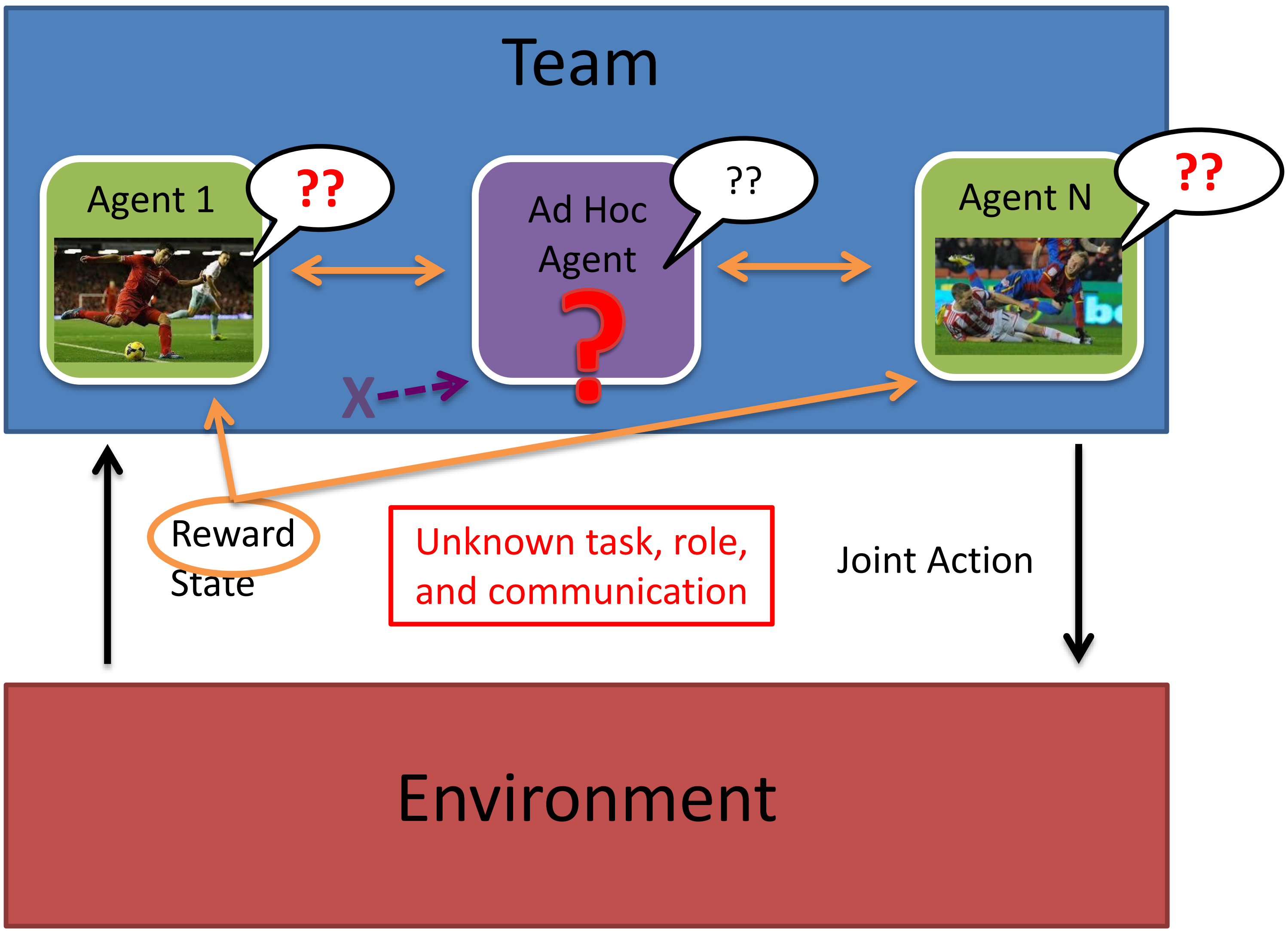
Ad Hoc Scenario



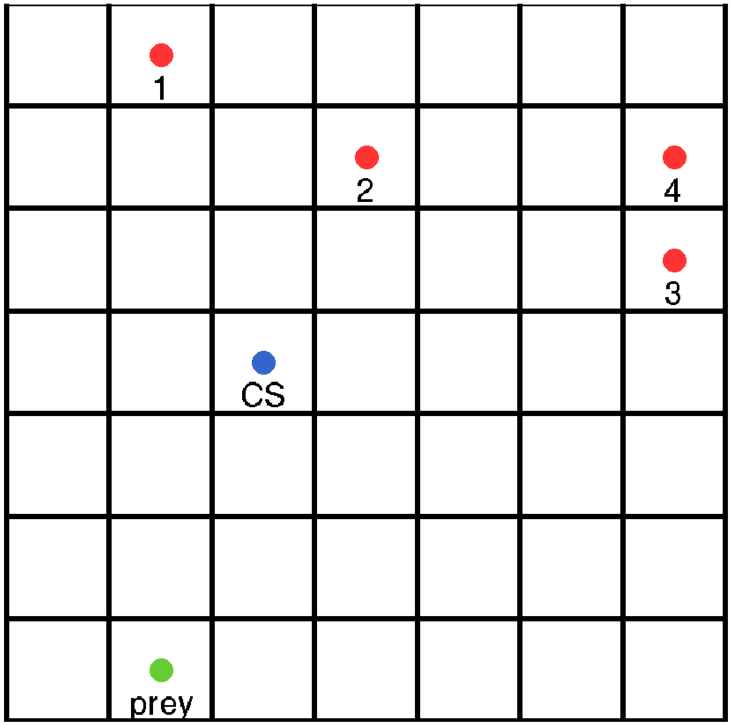
Ad Hoc Scenario



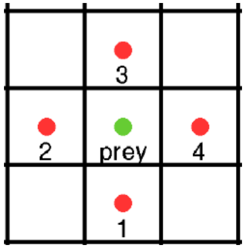
Ad Hoc Scenario



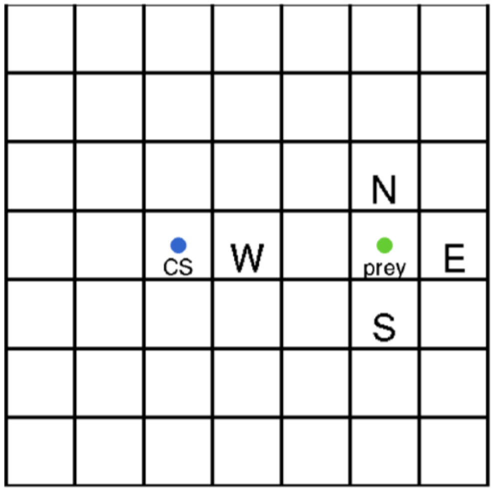
Pursuit Domain

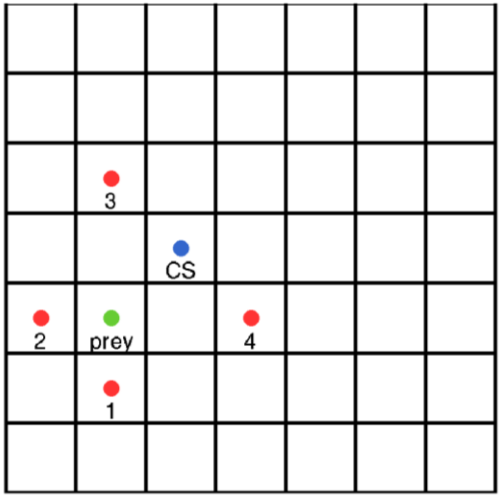


- 7x7 toroidal grid
- 4 agents, 1 prey
- Task: capture the prey at a specific position (CS)
- Prey move to an open neighbouring cell



Team strategy





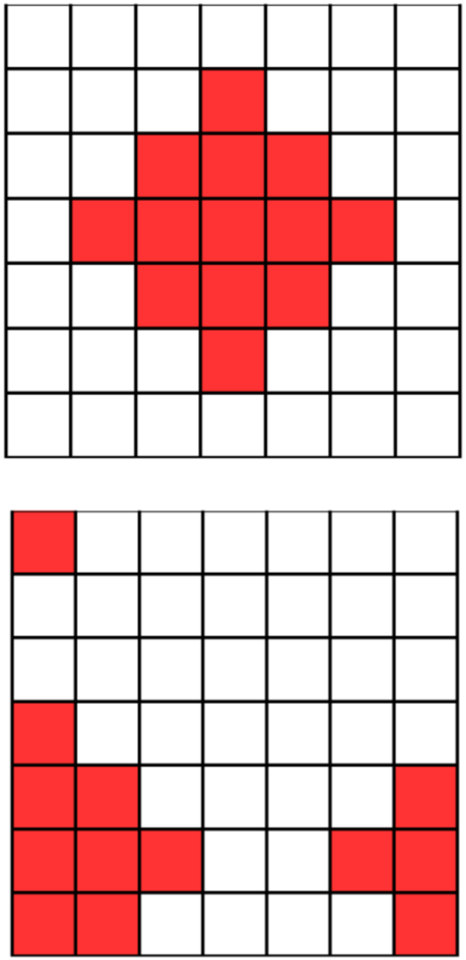
Agent are specialized to take the N, S, W, or E of the prey.

Constraining the prey to move in specific directions.

N,S,E,W depends on capture state, prey state, and position of agents.

N,S,E,W represented as reward. Agent policy computed independently using RL (Q-Learning)

Partial observability



Agents see 2 steps away

Might not see the prey -> prey state becomes probabilistic (impacts planning)

Communication

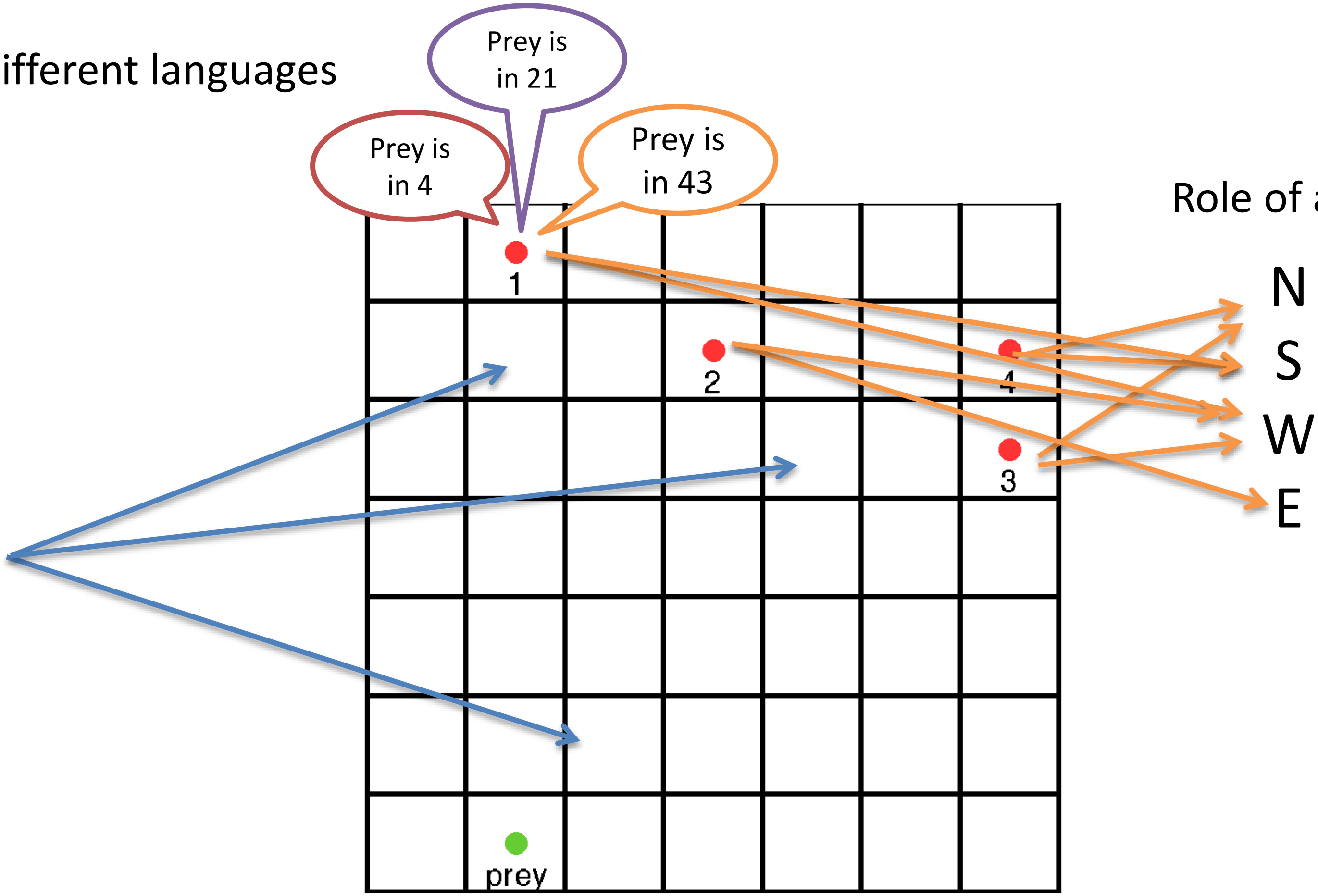
Agents communicate about prey position

Domain variables

Different languages

Role of agents

Capture state



10 possible capture state
10 possible languages
10 possible team configuration

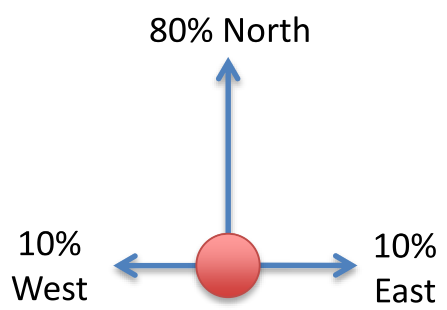
1000 hypothesis

Noise

Actions outcome are noisy

Communication is noisy

Impacts
planning and inference



Uniform probability to refer to a neighbouring cell

Method

A team **B** is made of **n_B** agents: **B = {b_1, …, b_nB}**

A domain **d** is made of:

- An environment **E** made of **n_S** states, and where agents can perform **n_U** actions. Environment dynamics is known as an MDP.
- A task **τ**, represented by a reward function **R**.
- A team configuration **K**, i.e. role given to each agent.
- A communication protocol **ρ**, i.e. agent language. We denote **m_b** as the message of an agent **b**.

A domain, **d={E, τ, K, ρ},** is a subset of all possible domains **D**.

Method

We assume a finite set of task, role, and communication protocol -> finite set of hypotheses **{d_1, .., d_h}**.

Ad Hoc can observe positions of all agents (**S and S'**) and their messages **M** (and sometimes the prey position)

We can use probabilistic update rules

Find the one that maximizes: $$\underset{h}{\operatorname{argmax}} \; p(d_h | S', S, M)$$

Method

Two sources of information

Messages

$$p(M|S, d_h)$$

Provides information on the communication system

$$p(S|M, S^{obs}, d_h)$$

Refines estimation of prey state (for partial observability)

Probability of next state given initial state and messages

$$p(S'|S, M, d_h)$$

Provides information on the task and roles

Method

$$p(S'|S, d_h) = \prod_i \sum_j p(s'_{b_i}|s_{b_i}, u_j, E_h) p(u_j|s_{b_i}, S, d_h)$$

Proba of next state given state for a given hypothesis

Over each agent

Over each action

Proba of next state for one agent, one action given the environment

Proba of action for this agent, given state of agents and domain hypothesis

Agents select their action independently

Method

$$p(d_h) = \sum_{s'_{prey}} \sum_{s_{prey}} p(d_h|S', S)p(s'_{prey})p(s_{prey})$$

Equation for full observability

Proba map of where the prey might be now

Proba map of where the prey might have been before

$$p(s_{prey}|M, s^{obs}_{prey}, S, d_h)$$
$$= \prod_i p(s_{prey}|m_{b_i}, s_{b_i}, s^{obs}_{prey}, \rho_h)$$

Updated given :

- message
- states of agents
- observed or not position of the prey
- hypotheses of communication protocol

Taking into account:

- knowledge of the prey policy
- noise in communication

Agents communicate independently

Planning

Ad Hoc agent weights policies of each hypothesis according to their probabilities

$$p(u_a|M,S) = \sum_h p(u_a|M,S,d_h)p(d_h)$$

Ad Hoc policy

Sum over all hypothesis

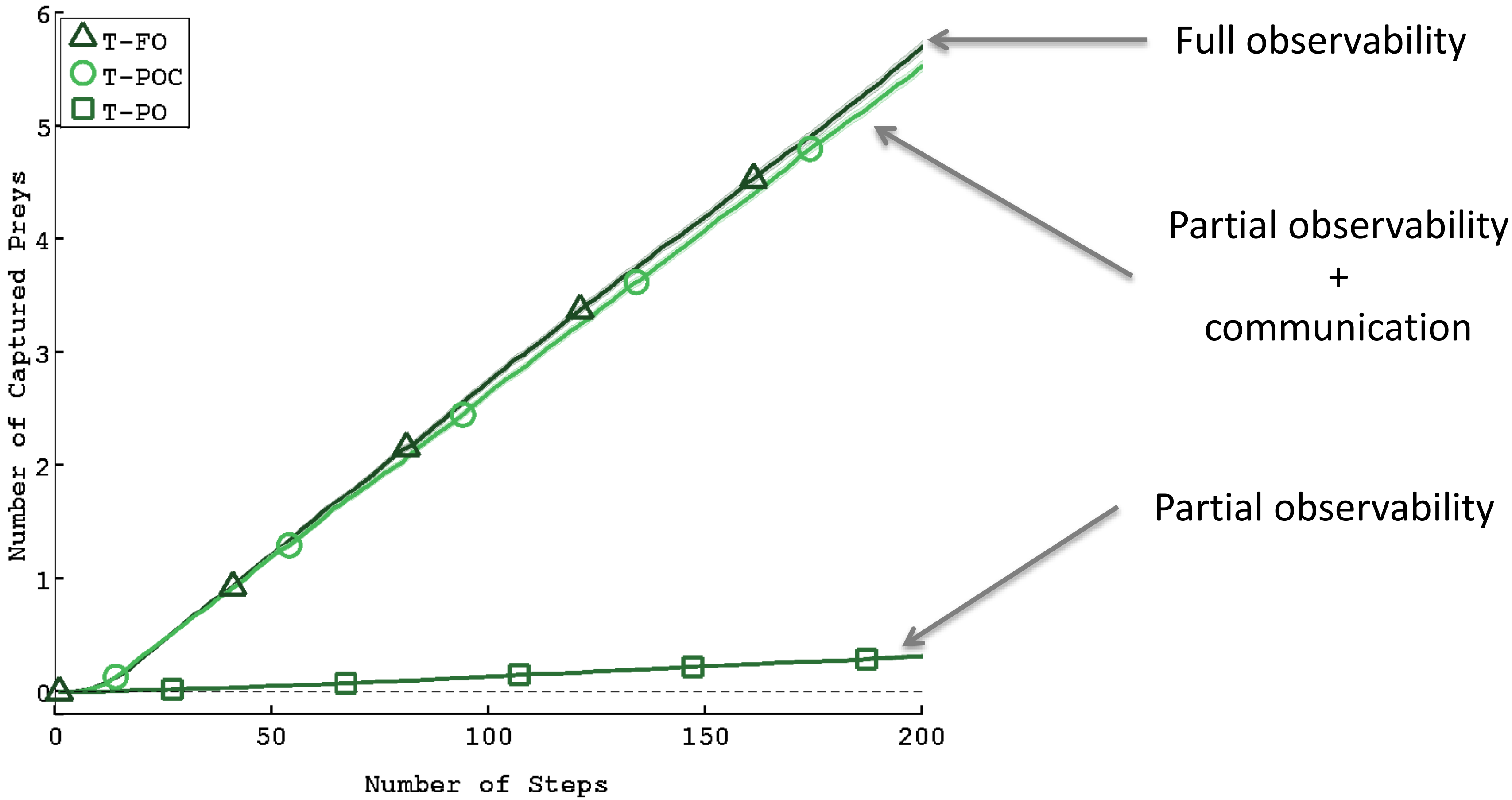
Hypothesis policy

Proba of hypothesis

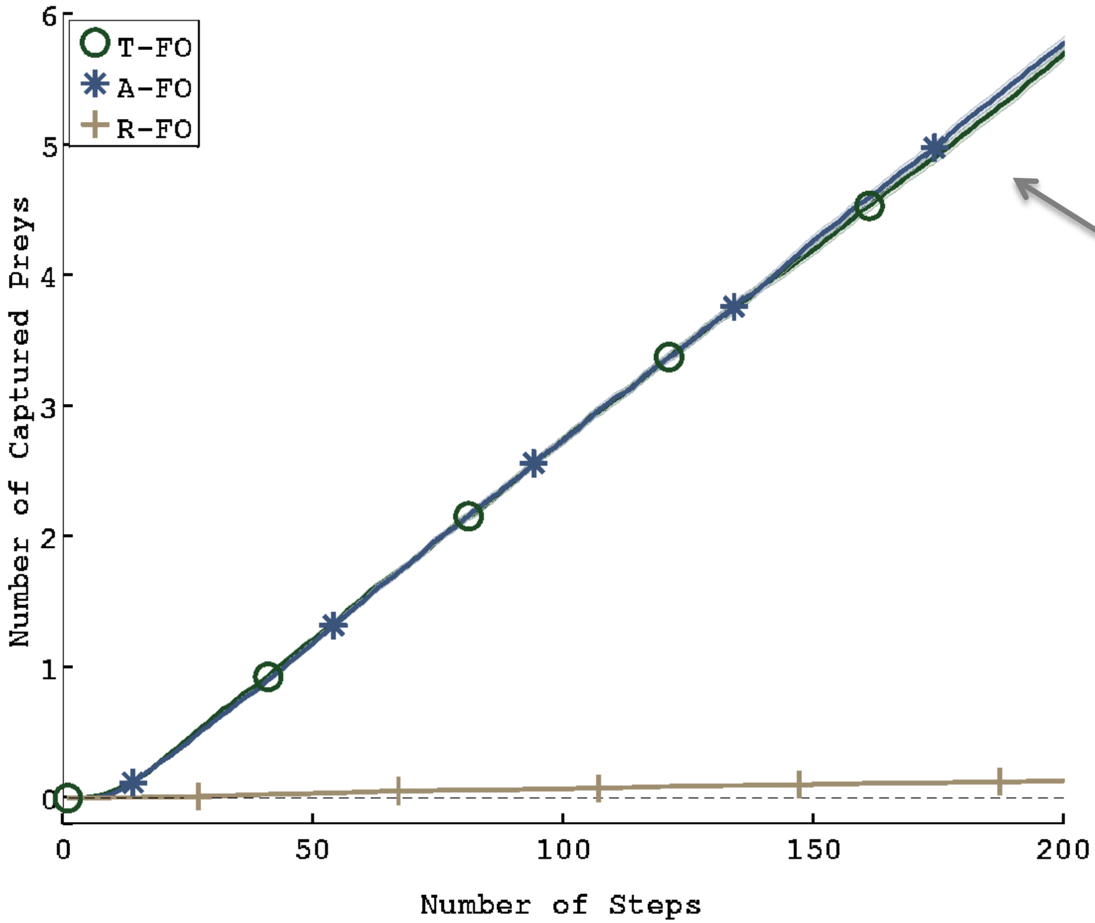
Evaluation

- Run for 200 steps
- Average over 1000 experiments
- Count the number of time a prey is captured
- Reset to random position after each capture
- Compare original team and ad hoc team

Default team

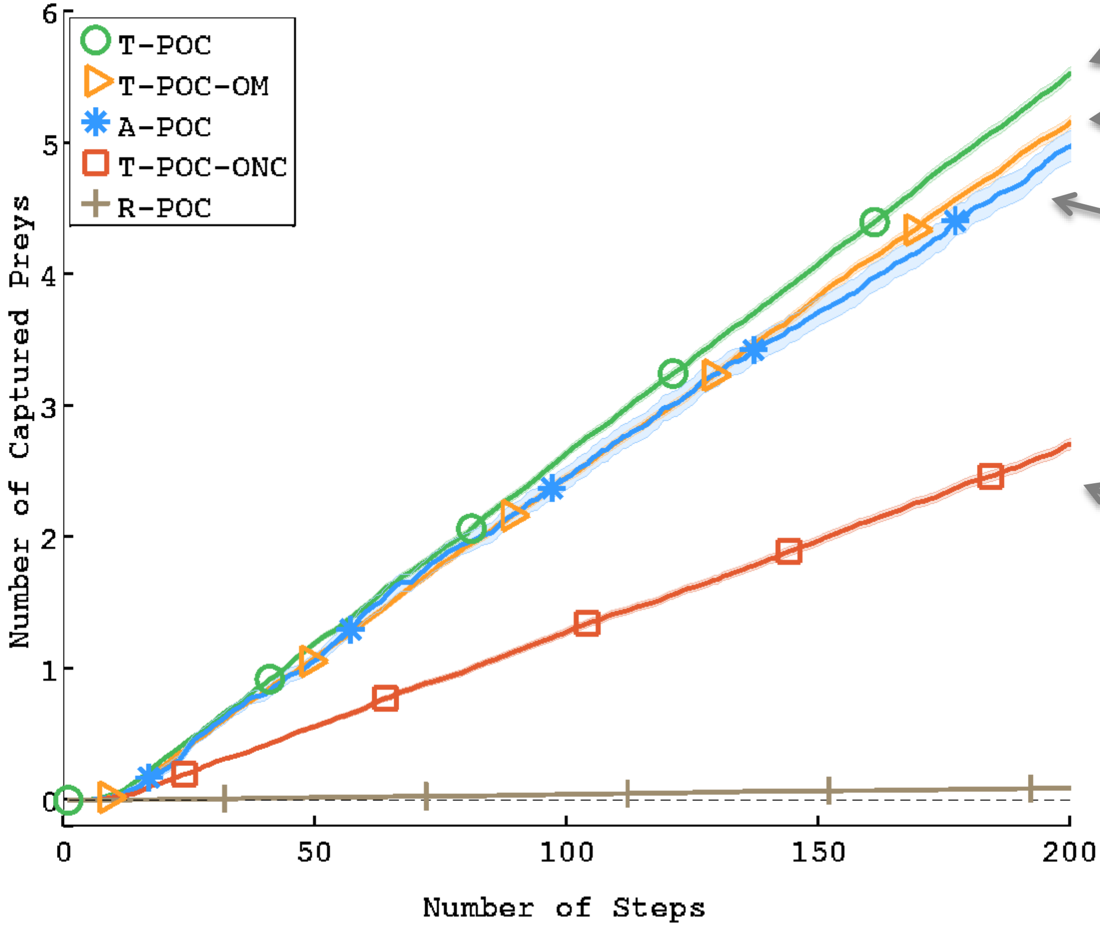


Full observability



Ad Hoc agent can match with the team seamlessly

Partial observability



Default team

Default team, one agent does not speak

With Ad Hoc agent
(does not speak but listen)

Default team, one agent neither speak nor listen

Conclusion

An Ad Hoc agent can integrate into a team without knowing the task, its role, and the communication protocol.

We assumed a finite and known set of possible task, role, and communication.

First time these three aspects are considered simultaneously in an ad hoc setting.

Limitations and future work

Finite (and relatively small) set of hypotheses

Yet huge computational time in the first few steps
(many many probable cases to evaluate)

Try sampling based method

Limitations and future work

Default team strategy is not optimal

Our adhoc agent can have similar performance despite not performing the same action in the first steps!

Use more advance planning methods for the team

Collaboration in Ad Hoc Teamwork: Ambiguous Tasks, Roles, and Communication

Jonathan Grizou, Samuel Barrett

Peter Stone, Manuel Lopes

Code available online at:
https://github.com/jgrizou/adhoc_com