MULTI-AGENT REINFORCEMENT LEARNING

Lesson 4: Real-World Complexities – From Theory to Practice

JULIA WANG

It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change.

OPENING ACT: CREATIVE APPLICATION SHOWCASE

Student Presentations

Let's see your most innovative and feasible MARL application designs!

Most Innovative Idea

Most Feasible Idea

RECAP & AGENDA

Previously On MARL...

- We explored the **dynamics** of learning.
  - Self-Play for robust autocurricula.
  - Centralized Critics (CTDE) to stabilize policy gradients.
  - Mean Field Theory to handle large populations.
- We assumed a "perfect" world: full observability, free communication, and well-behaved agents.

Today's Mission: Embrace the Mess

- We dive into real-world technical challenges.
  1. **Partial Observability**: Acting in a fog of war.
  2. **Communication**: The art and science of talking.
  3. **Robustness & Safety**: Surviving in a hostile world.

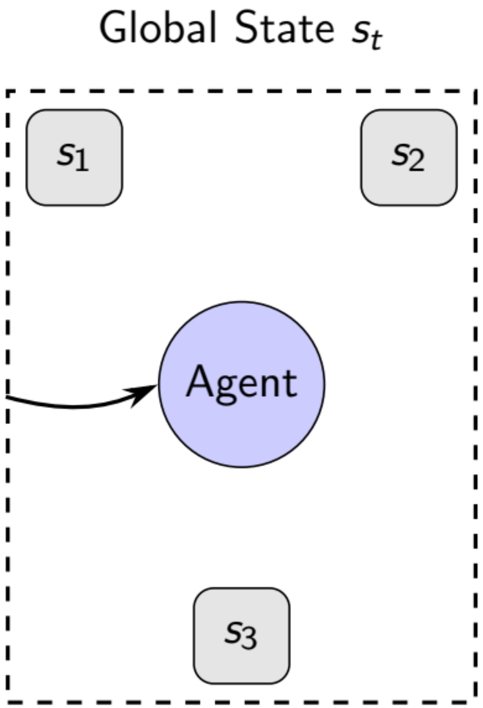
CHALLENGE 1: THE FOG OF WAR

Decision-Making with Incomplete Information

THE REALITY: FROM MDPS TO POMDPS

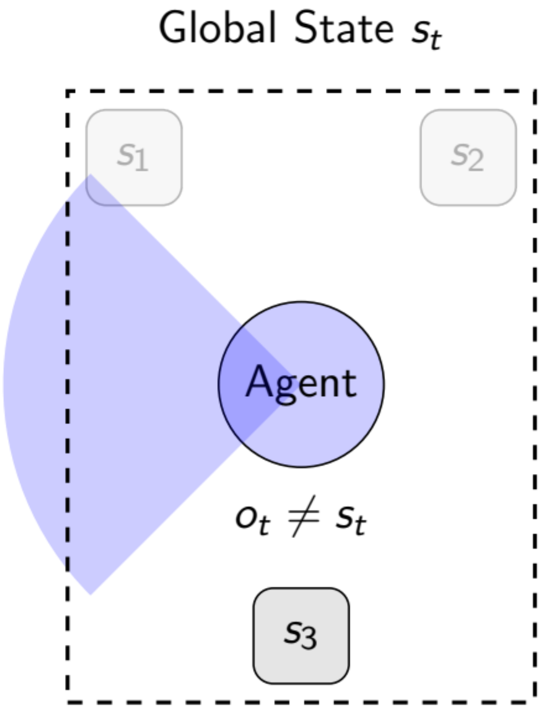
In an MDP, the agent knows the true state s_t. The real world is a **Partially Observable** MDP (POMDP).

Full Observability (MDP)



Analogy: Playing chess with a full view of the board.

Partial Observability (POMDP)

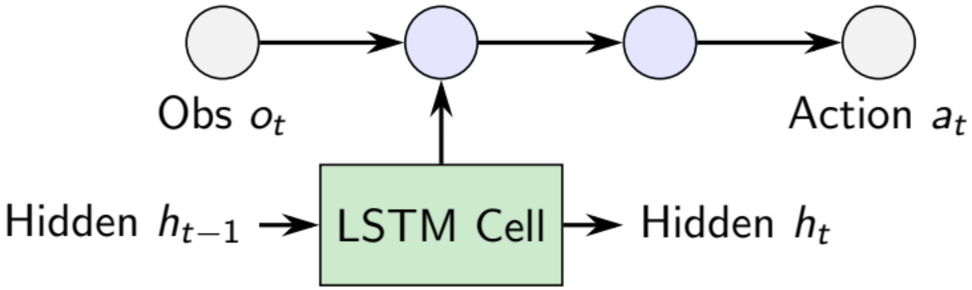


Analogy: Driving in thick fog.

SOLUTION 1: MEMORY – USING HISTORY TO BUILD STATE

If one observation is not enough, use a history of observations to infer the underlying state.

Idea: Integrate a memory module, like an LSTM or GRU, into the agent's policy network.



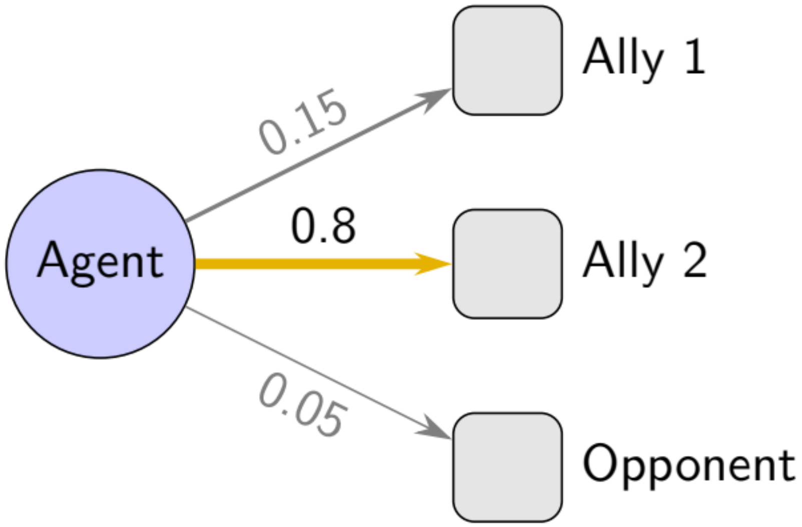
The agent's "belief" about the state b_t is a function of its previous belief and current observation: $b_t = f(b_{t-1}, o_t)$. An RNN is a natural way to implement this function f.

SOLUTION 2: ATTENTION – A SPOTLIGHT IN THE FOG

The Intuition: Not all information is equally important. Attention lets an agent dynamically focus on the most relevant parts of its observation.

In MARL: Who should I pay attention to?

- The teammate with the ball?
- The closest opponent?
- The manager giving orders?



The agent learns to assign high attention weights to relevant entities.

Discussion: Global vs. Local Attention

Should an agent attend to *all* other agents (expensive but complete), or only agents within a certain radius (cheaper but may miss critical long-range interactions)?

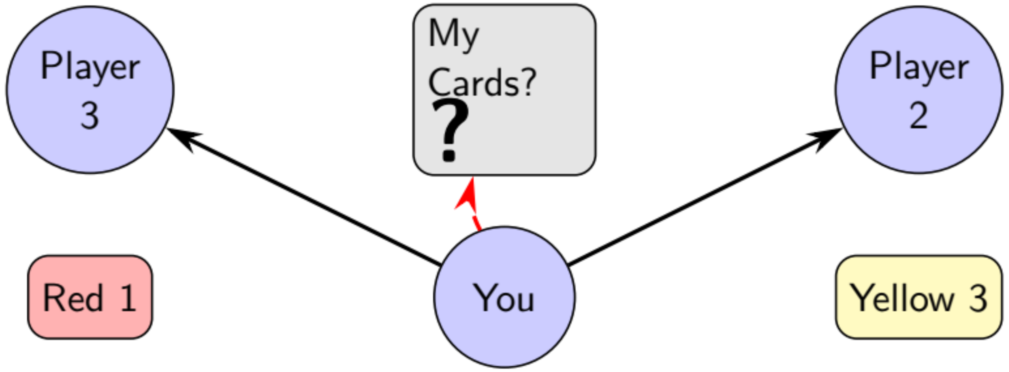
CASE STUDY: THE HANABI CARD GAME

Goal: A cooperative game where players must play cards in ascending order (1-5) for each color suit.

The Catch: Partial Observability

- You can see every other player's cards, but you **cannot see your own hand**.
- You must rely on hints from your teammates to infer what cards you hold.
- Hints are limited (e.g., "You have one red card," or "This card is a 4").

Inferring a Hidden State



Solution: Belief Modeling via Memory

An agent must use its memory (an RNN/LSTM) to integrate the history of all public actions and hints. It uses this history to build a "belief" (a probability distribution) over the cards it likely holds, allowing it to make an informed decision.

CHALLENGE 2: THE ART AND SCIENCE OF COMMUNICATION

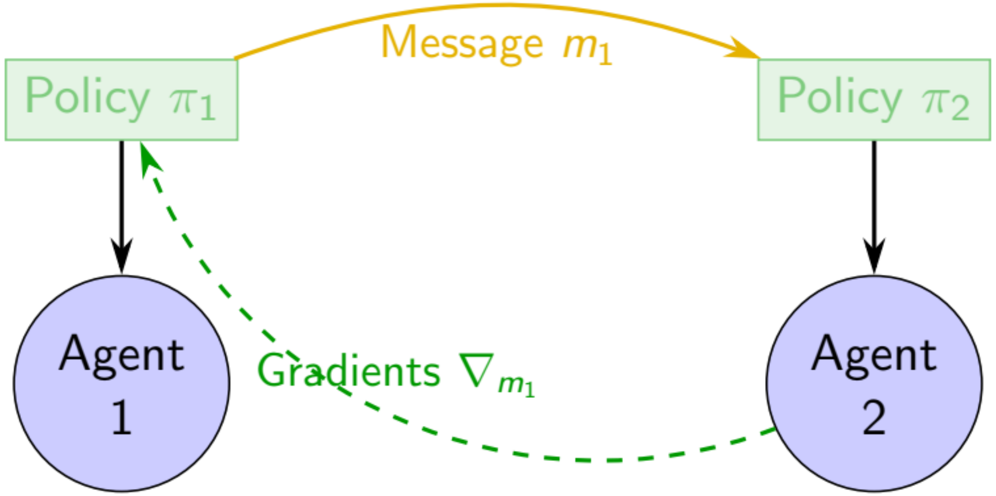
Can Agents Learn to Talk?

LEARNABLE COMMUNICATION: INVENTING A LANGUAGE

Problem: Can we let agents *learn* what to say?

Solution: Differentiable Communication (e.g., DIAL)

Make the communication channel part of the end-to-end learning process. Gradients flow from one agent's loss back through the channel to another agent's message-creation policy.



Side Effect: Emergent Languages

The resulting protocols are often highly efficient but completely alien to humans.

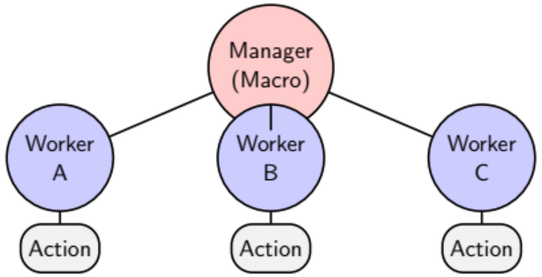
COMMUNICATION IS NOT FREE: EFFICIENCY AND HIERARCHY

Efficiency Trade-offs
Real-world systems have limited bandwidth and latency.
Solutions:

- **Selective Communication**: A gating mechanism learns *when* to send a message.
- **Message Compression**: Use autoencoders to compress information.

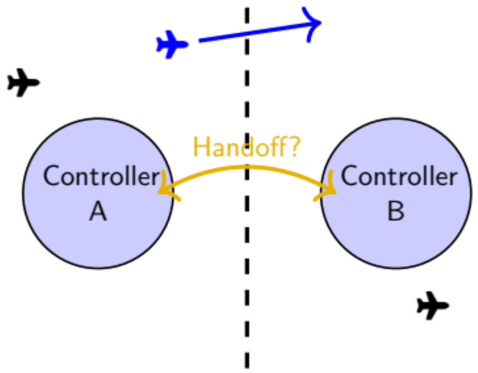
Hierarchical Coordination
Use a company-like structure instead of all-to-all communication.
Manager-Worker:

- A "Manager" agent sets high-level goals.
- "Worker" agents perform low-level actions to achieve them.



CASE STUDY: COLLABORATIVE AIR TRAFFIC CONTROL

Goal: A team of AI controllers must manage a crowded airspace, ensuring safety while minimizing flight delays.



Agents must communicate to hand off control of aircraft crossing sector boundaries.

The Communication Challenge

- Agents can't just broadcast their state. They need structured, targeted messages.
- What information is essential? The plane's ID, speed, heading, requested altitude?
- When is the best time to send a message to avoid ambiguity or distraction?

Solution: Learned Communication Protocol

By learning their own protocol, agents can develop a highly efficient language to coordinate complex actions like aircraft handoffs, achieving a level of collaboration that far surpasses hard-coded systems.

CHALLENGE 3: ROBUSTNESS AND SAFETY

Surviving in a Hostile and High-Stakes World

SURVIVING ADVERSARIES AND NOISE

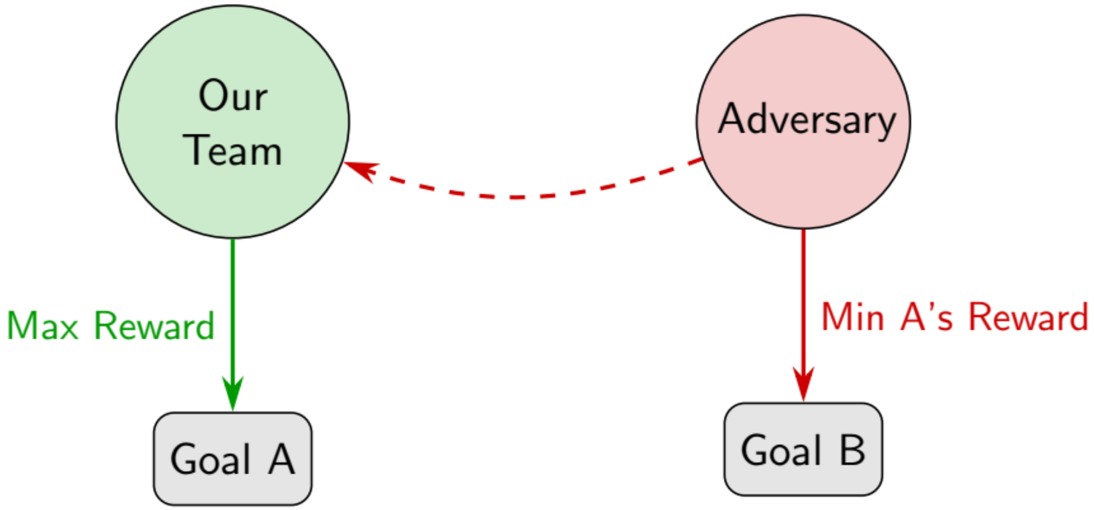
The real world is not always cooperative. We need agents robust to unexpected disturbances.

Threats

- **Noise**: Sensor errors, packet loss, action failures.
- **Adversaries**: Malicious agents actively trying to disrupt your team.

Defense Strategies

- **Domain Randomization**: Train with random noise in observations, actions, and physics.
- **Adversarial Training**: Train a second agent whose goal is to make your team fail. Your agents then learn to counter these worst-case disruptions.



Training against a dedicated adversary.

THE IMPORTANCE OF BEING SAFE

The Ultimate Challenge

In applications like autonomous driving, maximizing performance is secondary to **never causing catastrophic failure**.

Problem: A standard RL agent explores all actions, including unsafe ones, to find rewards.
Solution: Safety Layers / Shielding

- Train a high-performance, but potentially unsafe, RL policy.
- Use a formally verified, conservative "safety policy" alongside it.
- The safety layer acts as a shield, monitoring the RL agent's intended actions and overriding any that are deemed unsafe with a safe fallback.

CASE STUDY: SMART GRID CONTROL

Goal: A network of AI agents control power generators and distributors to meet fluctuating electricity demand, prevent blackouts, and minimize costs.

Hostile High-Stakes World

- **Noise**: Must handle unpredictable demand spikes (e.g., a heatwave) and sudden generator failures.
- **Adversaries**: Must be robust to potential cyberattacks trying to destabilize the grid.
- **Safety**: The ultimate constraint is maintaining grid frequency within a tiny, safe margin. Failure means a city-wide blackout.

Agents must balance the grid while defending against noise and attacks.

Solution: Layered Safety Adversarial Training

Agents are trained to be robust against failures and attacks, while a critical safety shield overrides any unsafe command to guarantee stability.

HOMEWORK: PRACTICAL CHALLENGE

Goal

- Solve a POMDP using an agent with memory.

Environment

- Modified PettingZoo simple_spread.
- Observation is missing velocity.

Your Task

- Start with the provided memory-less baseline agent.
- Implement a **DRQN** by adding an **LSTM layer** to the network.
- This LSTM acts as memory, allowing the agent to infer velocity from position history.
- Compare the performance of your DRQN vs. the baseline.

Core Challenge: Overcoming Partial Observability

KEY TAKEAWAYS

Your Journey Through Real-World MARL Challenges

Challenge 1: Partial Observability

The Problem:

- Agents can't see complete state
- Must infer from limited observations
- Like driving in fog

Solutions We Explored:

- **LSTM/GRU**: Memory networks to track history
- **Attention**: Focus on what matters
- **DRQN**: Deep Recurrent Q-Networks

Challenge 2: Communication

The Central Question:

- What, when, and how to communicate?

Solutions We Explored:

- **Differentiable Communication**: End-to-end learnable protocols
- **Selective Communication**: Send only when necessary
- **Hierarchical Coordination**: Manager-Worker architectures

NEXT TIME ON MARL...

Lesson 5: Cooperation, Competition, and Ad-Hoc Teamwork

We have seen how agents can learn and adapt. Now, we will explore the full spectrum of their interactions.

- **Fully Cooperative**: How do we solve the "credit assignment" problem? When the team gets one reward, who was responsible for the success? We'll look at methods like VDN and QMIX.
- **Fully Competitive**: Revisiting the "arms race" in zero-sum games, where one agent's win is another's loss.
- **Ad-Hoc Teamwork**: The ultimate challenge. How can your agent learn to collaborate with teammates it has *never seen before*? This is a critical step towards general-purpose AI assistants.

Questions?