Multi-agent Deep Reinforcement Learning

Learning with multiple agents

Maxime Toquebiau

VUB AI Research Group, Brussels

October 24th, 2025

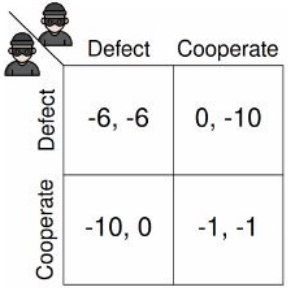
Table of Content

Introduction

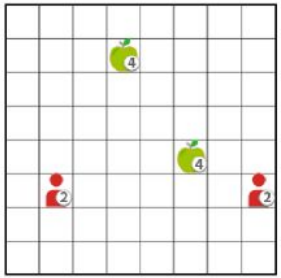
- Multi-agent systems
- From single to multi-agent systems
- Multi-agent deep reinforcement learning

Introduction

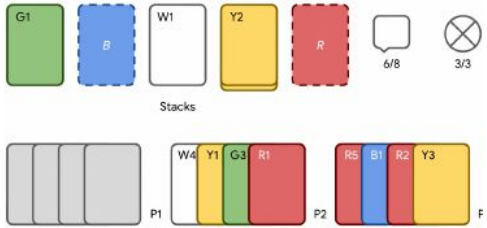
Multi-agent system



(a) Social dilemma



(b) Two-dimensional grid world



(c) Card game

(d) Multi-player video game

(e) Multi-robot system

(f) Human-robot interaction

Introduction

From single to multi-agent systems

Agents

Computer system that interacts with its environment, considered intelligent for being:

- ***reactive***: perceives information and acts in response to perceived signals
- ***proactive***: shapes its behaviour to satisfy its objectives
- ***learning***: learns from its experience

Multi-agent systems

A system of multiple intelligent agents that interact with their environment and with each other.

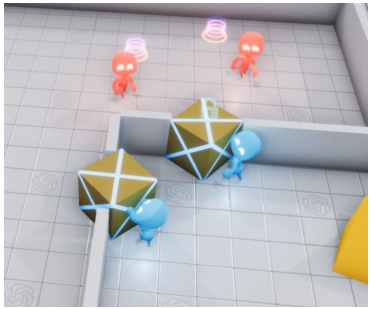
-> Agents are also:

- ***social***: shapes its behaviour depending on the behaviour of other agents

Introduction

Multi-agent deep reinforcement learning

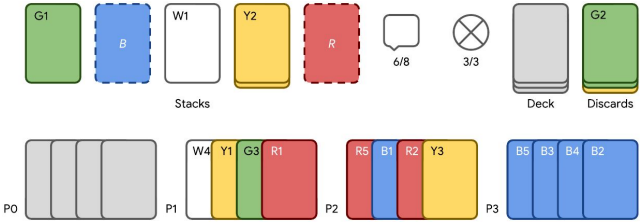
OpenAI Five (2019)[1]



Hide-and-seek (2019)[2]

Google Research Football (2019)[5]

Starcraft Multi-Agent Challenge (2019)[3]

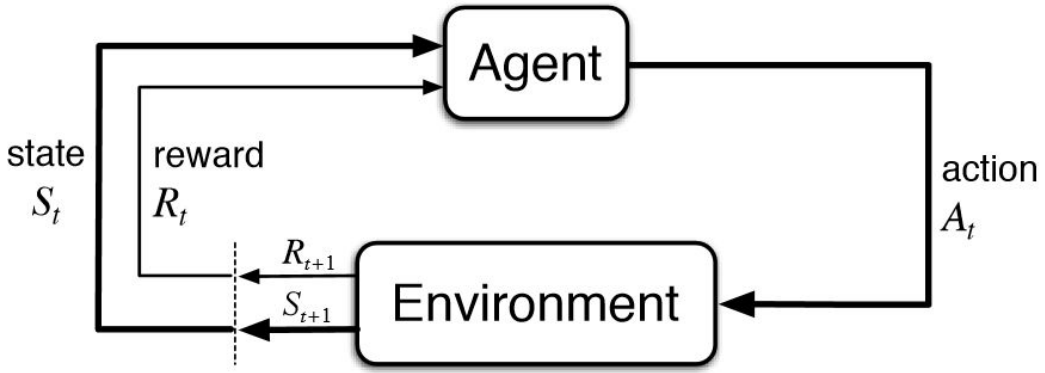


Hanabi (2019)[4]

[1]OpenAI et al., *Dota 2 with Large Scale Deep Reinforcement Learning*, 2019
[2]Baker et al., *Emergent Tool Use From Multi-Agent Autocurricula*, 2019
[3]Samvelyan et al., *The StarCraft Multi-Agent Challenge*, 2019
[4]Bard et al., *The Hanabi challenge: A new frontier for AI research*, 2020
[5]Kurach et al., *Google Research Football: A Novel Reinforcement Learning Environment*, 2019

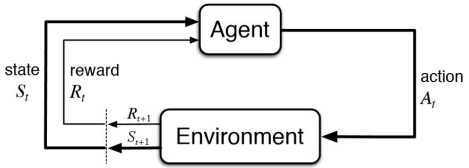
Introduction

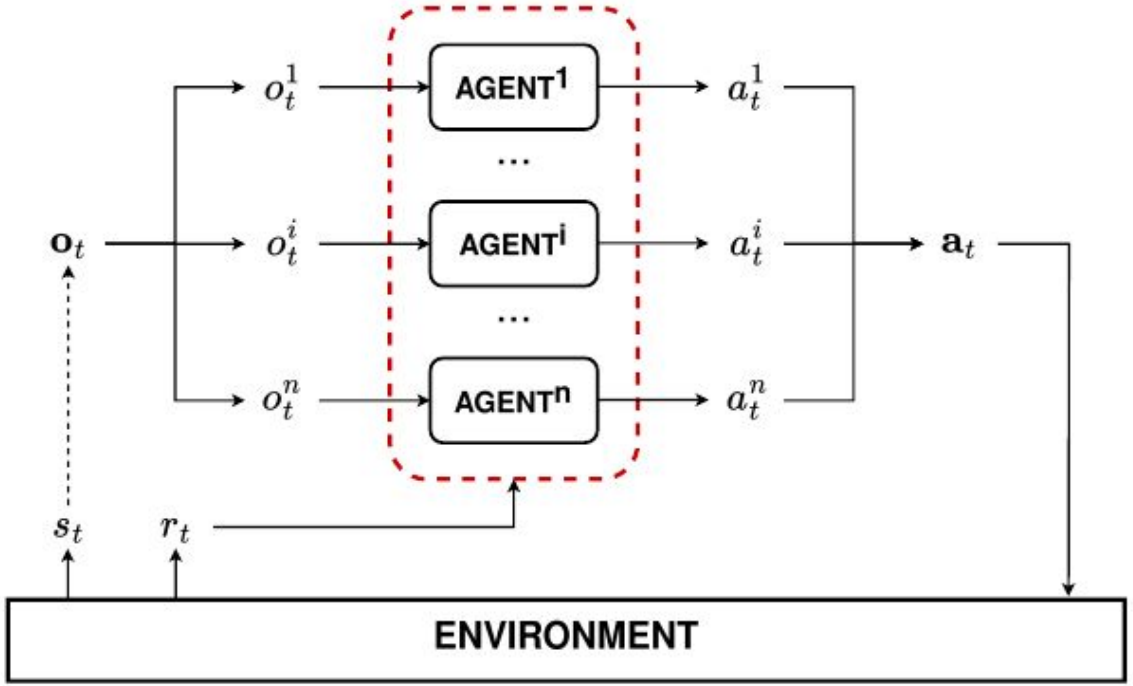
Multi-agent deep reinforcement learning



Introduction

Multi-agent deep reinforcement learning





Definitions

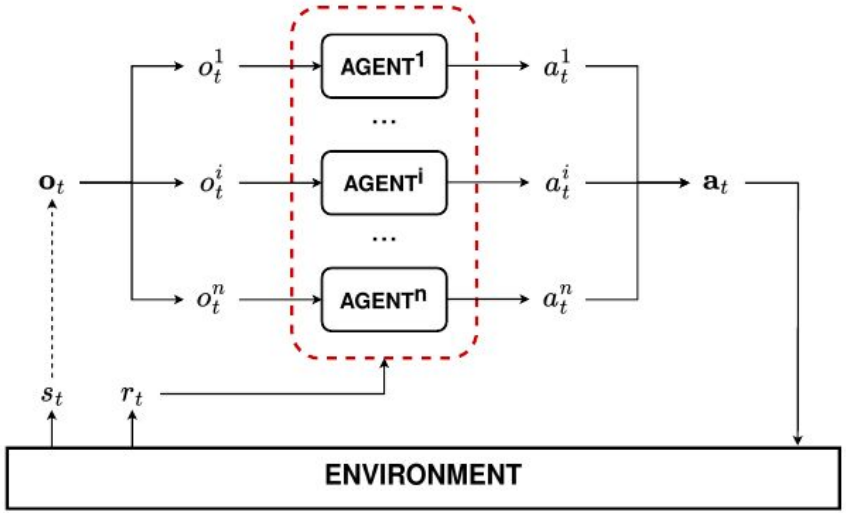
- Dec-POMDP
- Multi-agent RL tools
- Nash equilibrium
- Pareto optimality

Definition

Decentralised-Partially Observable MDP

$$\langle \mathbf{S}, \mathbf{A}, \mathcal{T}, \mathbf{O}, \mathcal{O}, \mathcal{R}, n, \gamma \rangle$$

- n number of agents
- $\mathbf{S}$ environment states, "**global states**"
- $\mathbf{O}$ set of **joint observations** $\mathbf{o} = \{o^1, ..., o^n\}$
- $\mathbf{A}$ set of **joint actions** $\mathbf{a} = \{a^1, ..., a^n\}$
- $\mathcal{T}$ transition function $P(s' | s, \mathbf{a})$
- $\mathcal{O}$ observation function $P(\mathbf{o} | \mathbf{a}, s')$
- $\mathcal{R}$ reward function
- γ discount factor



Oliehoek and Amato, *A concise introduction to decentralized POMDPs*, 2016

Definition

Multi-agent RL tools: joint and local policies

Joint Policy $\pi = (\pi^1, \ldots, \pi^n)$

Local policies $\pi^i(a^i|h^i) : \mathbf{H} \times \mathbf{A} \to [0, 1]$

history $h_t^i = (o_0^i, a_0^i, \ldots, a_{t-1}^i, o_t^i) \in \mathbf{H} = (\mathbf{O} \times \mathbf{A})^*$

Definition

Multi-agent RL tools: joint and local values

Joint value functions

$$V_\pi(\mathbf{h}_t) = \mathbb{E}_\pi[G_t \mid \mathbf{h}_t]$$
$$Q_\pi(\mathbf{h}_t, \mathbf{a}_t) = \mathbb{E}_\pi[G_t \mid \mathbf{h}_t, \mathbf{a}_t]$$

Local value functions

$$V_\pi(h_t^i) = \mathbb{E}_\pi[G_t \mid h_t^i]$$
$$Q_\pi(h_t^i, a_t^i) = \mathbb{E}_\pi[G_t \mid h_t^i, a_t^i]$$

Definition

Nash Equilibrium

A **Nash equilibrium** is a situation where no player could gain by changing their own strategy, *without harming the other players' gain*.

- Not necessarily unique

Pure-strategy equilibrium
→ deterministic joint policy

(A,A) or (B,B)

	A	B
A	1,1	1,-1
B	-1,1	0,0

Nash, *Equilibrium points in n-person games*, 1950

Definition

Nash Equilibrium

A **Nash equilibrium** is a situation where no player could gain by changing their own strategy, *without harming the other players' gain*.

Mixed equilibrium
→ stochastic joint policy

p(r)=⅓ p(p)=⅓ p(s)=⅓
for both players

1 \ 2	r	p	s
R	(0, 0)	(−1, 1)	(1, −1)
P	(1, −1)	(0, 0)	(−1, 1)
S	(−1, 1)	(1, −1)	(0, 0)

Nash, *Equilibrium points in n-person games*, 1950

Definition

Pareto optimality

An outcome of a game is **Pareto optimal** if there is no other outcome that makes every player at least as well off and at least one player strictly better off.

(A, A)

	A	B
A	1,1	1,-1
B	-1,1	0,0

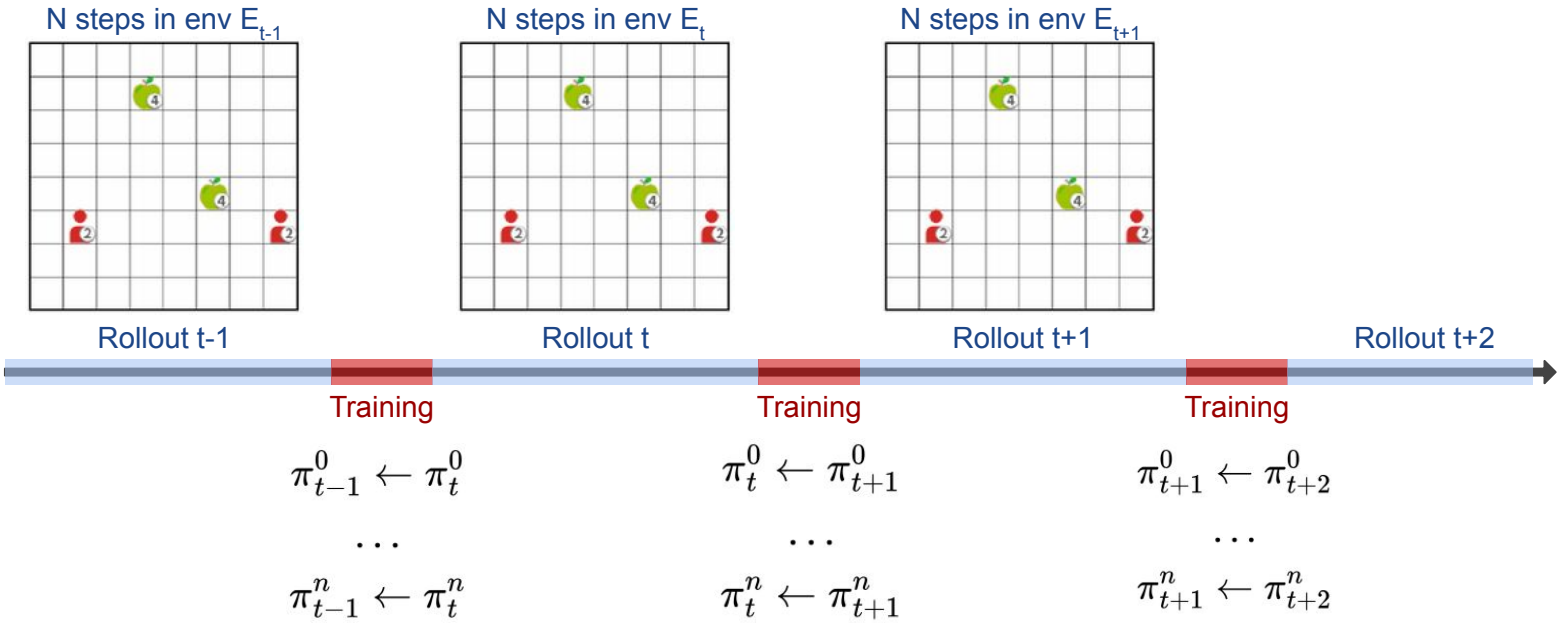
Learning with multiple agents

- Non-stationarity
- Partial observability
- Credit assignment
- Scaling

Learning with multiple agents

Non-stationarity

From a single agent's point of view, other agents are part of the environment.
→ If other agents change their strategy, the environment changes



Learning with multiple agents

Non-stationarity

From a single agent's point of view, other agents are part of the environment.
→ If other agents change their strategy, the environment changes

Problem: Experience replay is broken (Foerster2016)
Experience from old steps are not relevant because the agents have changed too much

→ **Concurrent experience replay** (Omidshafei2017)
Train all agents on the same environment steps, i.e., similar data will help agents learn similar things

→ **Weight down old experiences** (Foerster2017)

→ **Use smaller replay memory**

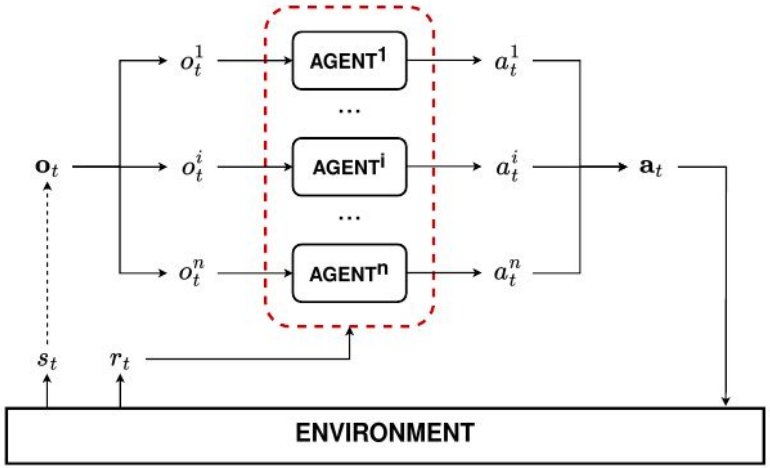
Foerster et al., *Learning to Communicate to Solve Riddles with Deep Distributed Recurrent Q-Networks*, 2016
Omidshafiei et al., *Deep Decentralized Multi-task Multi-Agent Reinforcement Learning under Partial Observability, 2017*
Foerster et al., *Stabilising Experience Replay for Deep Multi-Agent Reinforcement Learning*, 2017

Learning with multiple agents

Partial observability

Agents gather only **incomplete information about the environment**, i.e., the local observation resulting from sensors.

$$\mathcal{O}(\mathbf{a}, s') = P(\mathbf{o}|\mathbf{a}, s')$$
$$\mathbf{o} = \{o^1, \ldots, o^n\}$$



Each agent has a different local observation…
→ agents have asymmetrical knowledge of the environment

Learning with multiple agents

Partial observability

Agents gather only **incomplete information about the environment**, i.e., the local observation resulting from sensors.

Problem: Agents need to behave according to incomplete knowledge of the current state

→ **Beliefs?** difficult as the state depends on other agents (Oliehoek2016)

→ **Memory** (Hausknecht2015)
Memorise previously gathered information for better estimating the current state

→ **Communication** (Zhu2024)
Share local information with other agents to cooperatively build better state estimation

Oliehoek and Amato, *A concise introduction to decentralized POMDPs*, 2016
Hausknecht and Stone, *Deep Recurrent Q-Learning for Partially Observable MDPs*, 2015
Zhu et al., *A survey of multi-agent deep reinforcement learning with communication*, 2024

Learning with multiple agents

Credit assignment

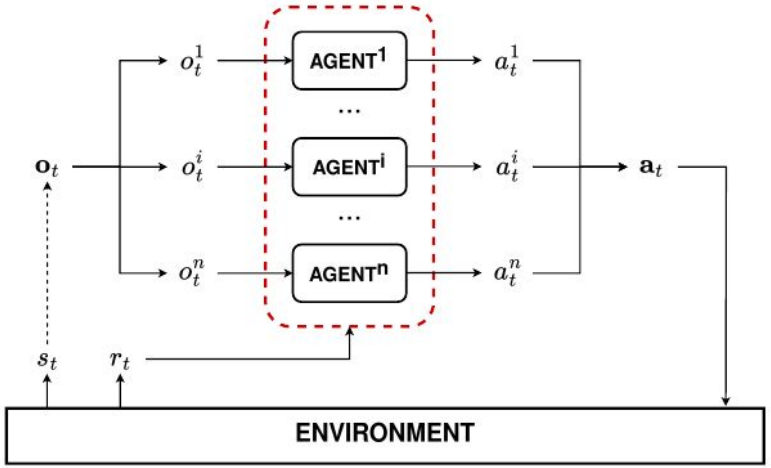
"Did I succeed (or fail) because of my actions, or because of the actions of other agents?"

i.e.,

How to measure the contribution of each agent's actions?

$$\mathcal{R} : \mathbf{S} \times \mathbf{A} \to \mathbb{R},$$

$$\mathcal{R}(s_t, \mathbf{a}_t) = r_{t+1}$$



Learning with multiple agents

Credit assignment

Credit assignment aims to find the real **marginal contribution** of each action.

$$r_{t+1} = \sum_i u^i_{t+1}, \text{ with } u^i_{t+1} = u(a_t)$$

→ **Shapley value** (Shapley1953) $u^i = \mathbb{E}_{C \subset N \setminus \{i\}} \left[u(C \cup i) - u(C) \right]$

i.e., contribution to any possible coalition

→ **Wonderful Life Utility** (Wolpert1999) $u^i = u(\mathbf{a}) - u(\mathbf{a}_{-i}, a_i = \text{null})$

i.e., contribution compared to counterfactual baseline

→ **Aristocrat Utility** (Wolpert2002) $u^i = u(\mathbf{a}) - \mathbb{E}[u|\mathbf{a}_{-i}, s]$

Shapley, *A value for n-person games*, 1953
Wolpert and Tumer, *An Introduction to Collective Intelligence*, 1999
Wolpert and Tumer, *Optimal Payoff Functions for Members of Collectives*, 2002

Learning with multiple agents

Scaling

How to learn in larger environments, ***with more agents ?***

Problems:

- worse non-stationarity
- harder credit assignment
- harder joint policy learning

→ **Deep learning?**

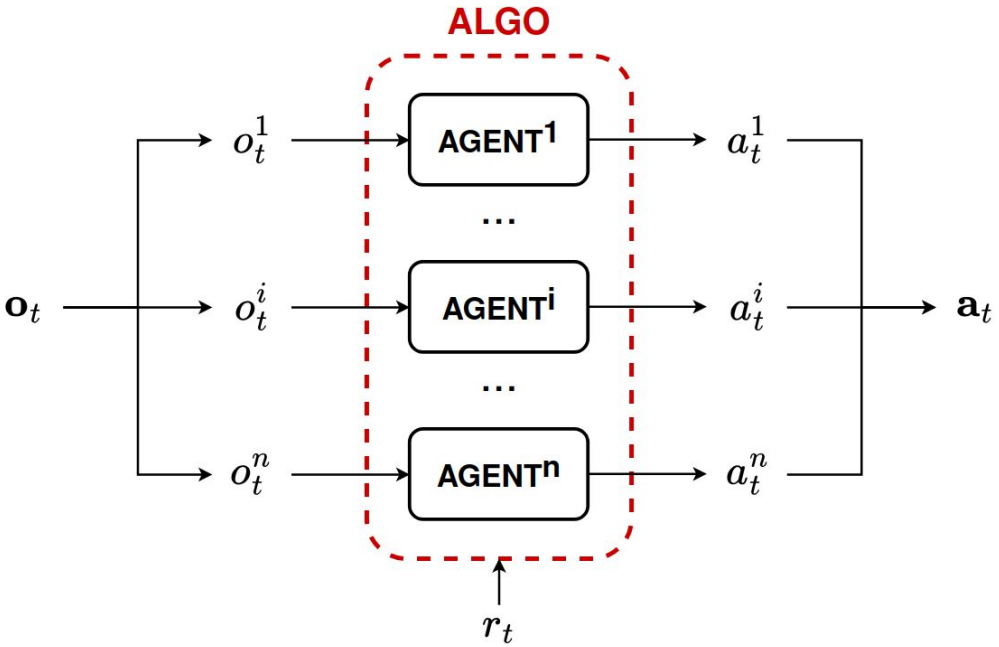
Multi-agent deep reinforcement learning

- Multi-agent learning paradigms
- Independent learning
- Multi-agent actor-critics
- Value factorisation
- Emergent communication
- Agent modelling

Multi-agent (deep) reinforcement learning

Learning paradigms

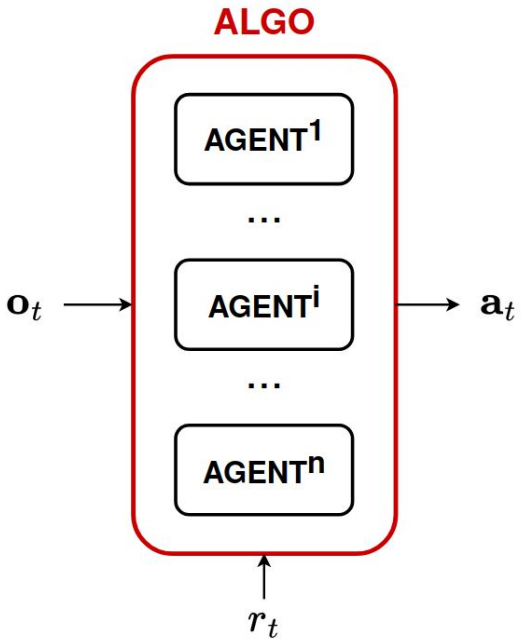
How to define the learning algorithm?



Multi-agent (deep) reinforcement learning

Learning paradigms: Reducing to single-agent RL

Fully centralised



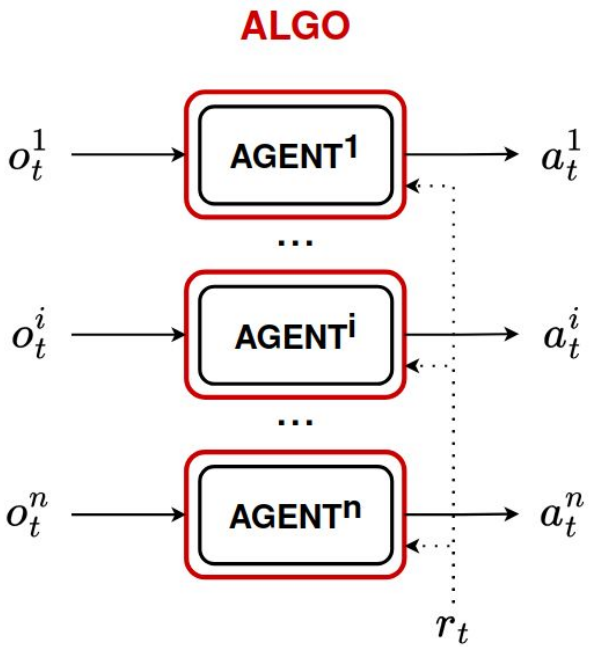
+ No more non-stationarity
+ No more credit assignment
+ Better partial observability

- Worse scaling ability:
  - larger observation space
  - larger action space
- Not possible in realistic environments

Multi-agent (deep) reinforcement learning

Learning paradigms: Reducing to single-agent RL

Fully decentralised



\+ Better scaling

\+ More compute-efficient

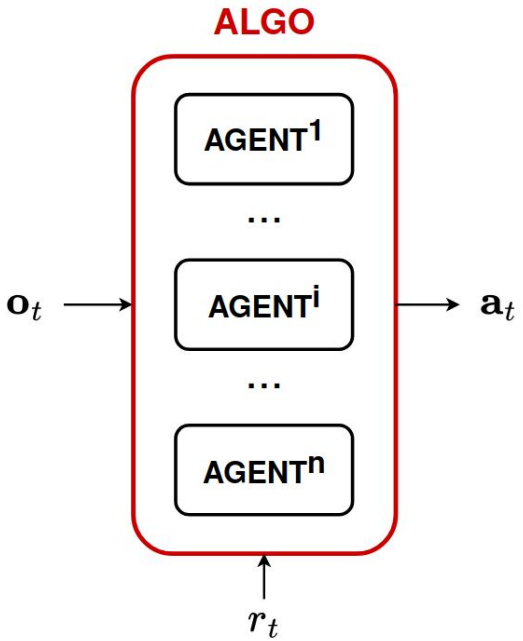
\- Non-stationarity

\- Credit assignment

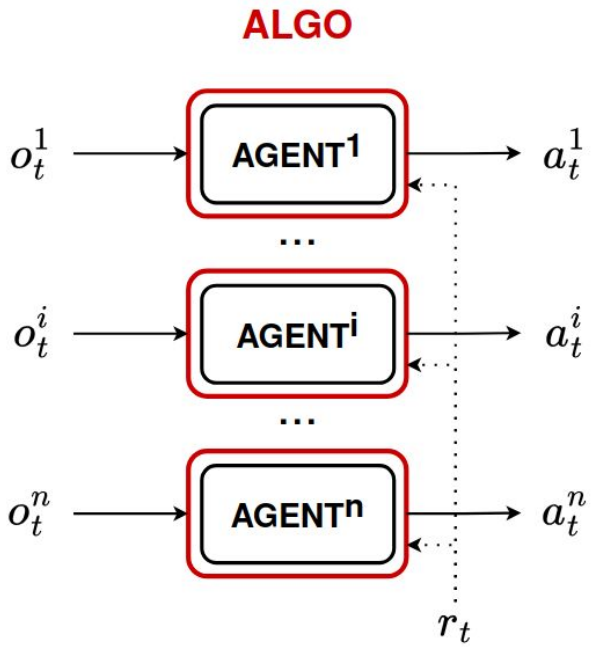
Multi-agent (deep) reinforcement learning

Learning paradigms: Reducing to single-agent RL

Fully centralised



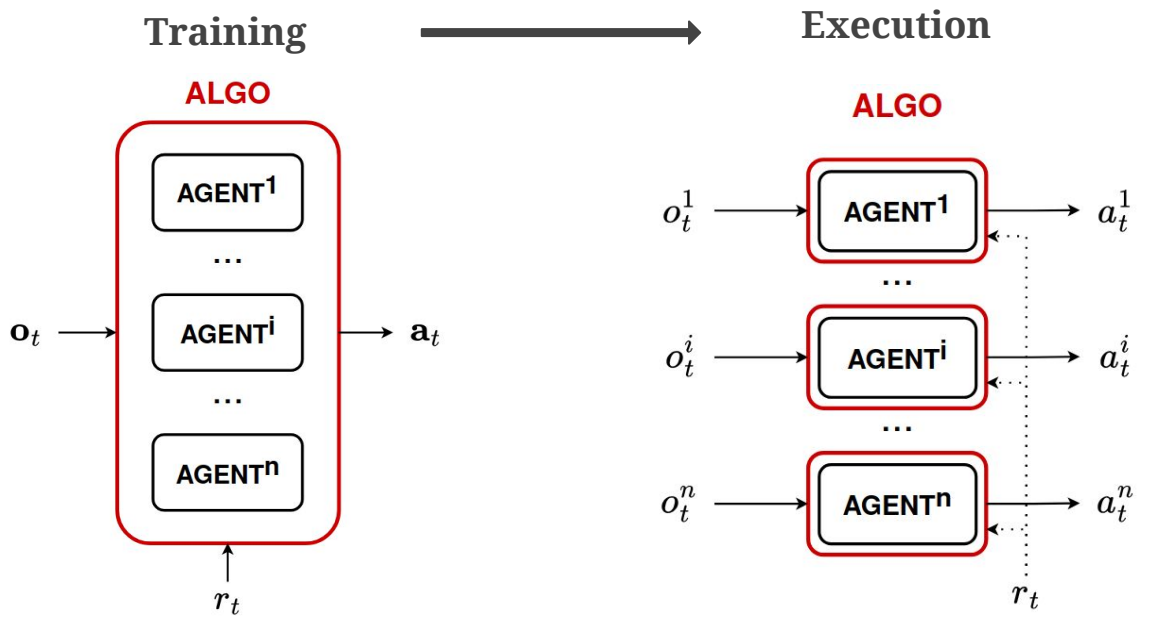
Fully decentralised



Multi-agent (deep) reinforcement learning

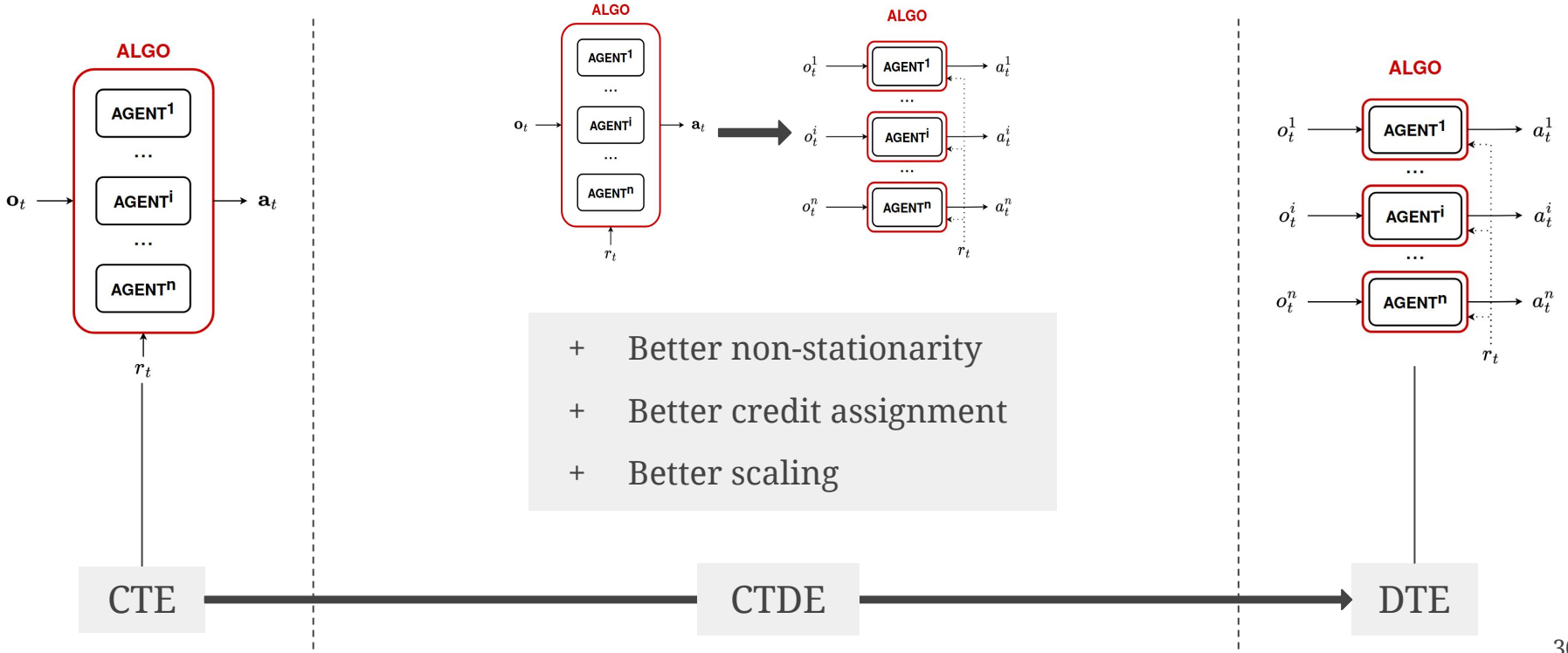
Learning paradigms: Training/Execution

Centralised Training with Decentralised Execution



Multi-agent (deep) reinforcement learning

Learning paradigms: Centralised Training with Decentralised Execution



Multi-agent (deep) reinforcement learning

Independent Learning

Independent Learning: Considering each agent independently… with some simplifications. (see Jiang2024)

→ **Independent Q-learning** (Tan1993)
- sharing parameters between agents

→ **Concurrent experience replay** (Omidshafei2017)

→ **Population-based training** (Jaderberg2019)
- training large number of agents, with changing teams → better generalisation

→ **Ideal Independent Q-learning** (Jiang2022)
- Consider other agents will act optimally

Tan, *Multi-Agent Reinforcement Learning: Independent versus Cooperative Agents*, 1993
Omidshafiei et al., *Deep Decentralized Multi-task Multi-Agent Reinforcement Learning under Partial Observability, 2017*
Jaderberg et al., *Human-level performance in first-person multiplayer games with population-based deep reinforcement learning*, 2019
Jiang and Lu, *I2Q: A Fully Decentralized Q-Learning Algorithm*, 2022
Jiang et al., *Fully Decentralized Cooperative Multi-Agent Reinforcement Learning: A Survey*, 2024

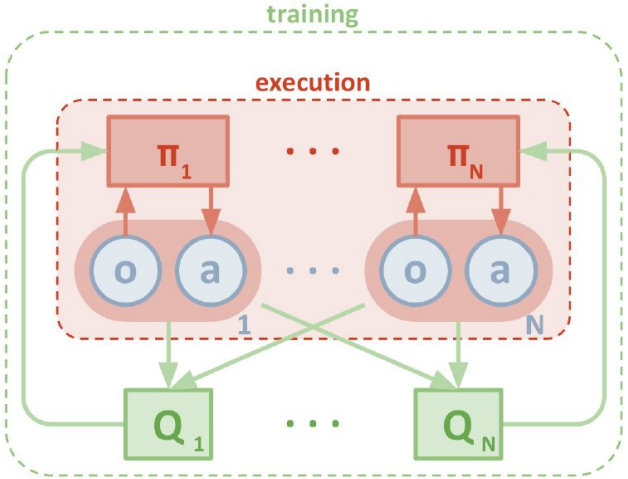
Multi-agent (deep) reinforcement learning

Multi-agent actor-critics

To fit the CTDE paradigm, agents only need to have decentralised policies.
→ value function can be centralised, as they're used only during training

→ **Multi-Agent DDPG** (Lowe2017)

- decentralised policies, trained by centralised critics
- one policy per agent $\pi_i(o^i) = a^i$
- one critic per agent $Q_i(\mathbf{o}, a^i)$



Lowe et al., *Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments*, 2017

Multi-agent (deep) reinforcement learning

Multi-agent actor-critics

To fit the CTDE paradigm, agents only need to have decentralised policies.
→ value function can be centralised, as they're used only during training

→ **Multi-Agent TD3** (Ackermann2019)

→ **Multi-Agent SAC** (Yu2021a)

→ **Multi-Agent PPO** (Yu2021b)

→ **COMA** (Foerster2018)

- Aristocrat utility computed with a centralised critic:

$$A^i(\mathbf{h}, \mathbf{a}) = Q(\mathbf{h}, \mathbf{a}) - \sum_{a^{i\prime}} \pi^i(a^{i\prime}|h^i) Q(\mathbf{h}, (\mathbf{a}^{-i}, a^{i\prime}))$$

Ackermann et al., *Reducing Overestimation Bias in Multi-Agent Domains Using Double Centralized Critics*, 2019
Yu et al., *Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms,* 2021a
Yu et al., *The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games*, 2021b
Foerster et al., *Counterfactual Multi-Agent Policy Gradients*, 2018

Multi-agent (deep) reinforcement learning

Value factorisation

How can we learn the value of local actions?

$$Q_\pi(h_t^i, a_t^i) = \mathbb{E}_\pi[G_t \mid h_t^i, a_t^i]$$

Problem: This value uses the return G_t, but G_t depends on the actions of other agents…

To use Q-learning for action-selection, we may instead need:

$$Q_{\pi^i}(h^i, a^i) = \mathbb{E}_{\pi^i}\left[U_t^i | h_t^i, a_t^i\right],$$

$$\text{with, } U_t^i = \sum_{k=t}^{T} \gamma^{k-t} u_k^i$$

Multi-agent (deep) reinforcement learning

Value factorisation

How can we learn the value of local actions?

$$Q_{\pi^i}(h^i, a^i) = \mathbb{E}_{\pi^i}\left[U_t^i | h_t^i, a_t^i\right]$$

This is a **credit assignment issue**, i.e., we need to find how these local values compose the global on that estimates the return:

$$Q(\mathbf{h}, \mathbf{a}) = f\left(\{Q^i(h^i, a^i)\}_{1 \leq i \leq n}, \mathbf{h}\right)$$

$$Q_\pi(\mathbf{h}_t, \mathbf{a}_t) = \mathbb{E}_\pi[G_t \mid \mathbf{h}_t, \mathbf{a}_t]$$

Multi-agent (deep) reinforcement learning

Value factorisation

How can we compose local action-value into the global one?
i.e., we need to find a function *f*

$$Q(\mathbf{h}, \mathbf{a}) = f\left(\{Q^i(h^i, a^i)\}_{1 \leq i \leq n}, \mathbf{h}\right)$$

$$Q_{\pi^i}(h^i, a^i) = \mathbb{E}_{\pi^i}\left[U_t^i | h_t^i, a_t^i\right] \qquad Q_\pi(\mathbf{h}_t, \mathbf{a}_t) = \mathbb{E}_\pi[G_t \mid \mathbf{h}_t, \mathbf{a}_t]$$

→ To select local actions with Q^i, we need to guarantee the **Individual-Global-Max property**:

$$argmax_{\mathbf{a}} Q(\mathbf{h}, \mathbf{a}) = \{argmax_{a^i} Q^i(h^i, a^i)\}_{1 \leq i \leq n}$$

→ function *f* must guarantee the IGM

Rashid et al., *QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning*, 2018

Multi-agent (deep) reinforcement learning

Value factorisation

How can we compose local action-value into the global one?
i.e., we need to find a function *f* **that guarantees IGM**

IGM : $argmax_{\mathbf{a}} Q(\mathbf{h}, \mathbf{a}) = \{argmax_{a^i} Q^i(h^i, a^i)\}_{1 \leq i \leq n}$

→ **Value Decomposition Networks (VDN)** (Sunehag2018)

$$Q(\mathbf{h}, \mathbf{a}) \approx \sum_{i=1}^{n} Q^i(h^i, a^i)$$

- One DQN Q^i for each agent
- All local DQNs learnt from the MSBE of the global Q:

$$L(\theta) = \mathbb{E}\left[\left(r + \gamma \max_{\mathbf{a}'} Q(\mathbf{h}', \mathbf{a}') - Q(\mathbf{h}, \mathbf{a})\right)^2\right]$$

Sunehag et al., *Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward*, 2018

Multi-agent (deep) reinforcement learning

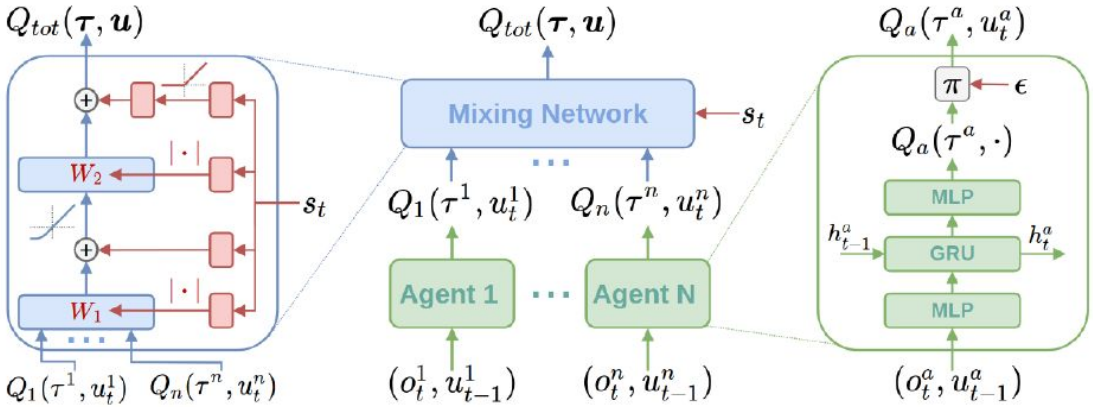
Value factorisation

How can we compose local action-value into the global one?
i.e., we need to find a function *f* **that guarantees IGM**

IGM : $argmax_{\mathbf{a}} Q(\mathbf{h}, \mathbf{a}) = \{argmax_{a^i} Q^i(h^i, a^i)\}_{1 \leq i \leq n}$

→ **QMIX** (Rashid2018)

- *f* modeled as neural network
- *f* takes the state as input, i.e., **the mixing strategy depends on the state**
- ***f*'s weights are made positive** to ensure the IGM property



Rashid et al., *QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning*, 2018

Multi-agent (deep) reinforcement learning

Value factorisation: QMIX extension

- → **VDN** (Sunehag2018)
- → **QMIX** (Rashid2018)
- → **QTRAN** (Son2019)
- → **Weighted QMIX** (Rashid2020)
- → **QPLEX** (Wang2021): IGM based on Advantage function instead of Q-value
- → **Qatten** (Yang2020): attention in mixing network
- → **LICA** (Zhou2020) & **FACMAC** (Peng2021): QMIX in actor-critic framework
- → **DFAC** (Sun2023): QMIX adapted to distributional DQN

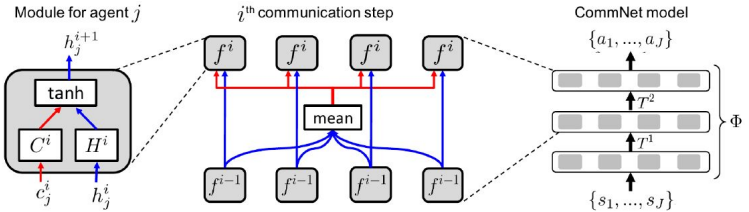
Sunehag et al., *Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward*, 2018
Rashid et al., *QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning*, 2018
Son et al., *QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning*, 2019
Rashid et al., *Weighted QMIX: Expanding Monotonic Value Function Factorisation for Deep MultiAgent Reinforcement Learning*, 2020
Wang et al., *QPLEX: Duplex Dueling Multi-Agent Q-Learning*, 2021
Yang et al., *Qatten: A General Framework for Cooperative Multiagent Reinforcement Learning*, 2020
Zhou et al., *Learning Implicit Credit Assignment for Cooperative Multi-Agent Reinforcement Learning*, 2020
Peng et al., *FACMAC: Factored Multi-Agent Centralised Policy Gradients*, 2021
Sun et al., *A Unified Framework for Factorizing Distributional Value Functions for Multi-Agent Reinforcement Learning*, 2023

Multi-agent (deep) reinforcement learning

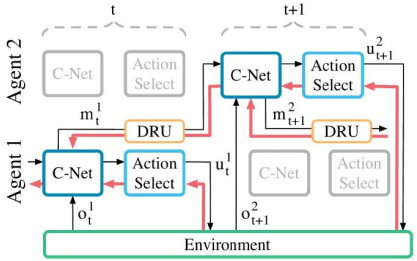
Emergent communication

Agents can learn to communicate to maximise their return, i.e., **emergent communication** (Lazaridou2020)

→ **CommNet** (Sukhbaatar2016)



→ **DIAL** (Foerster2016)



Sukhbaatar et al., *Learning Multiagent Communication with Backpropagation*, 2016
Foerster et al., *Learning to Communicate with Deep Multi-Agent Reinforcement Learning*, 2016
Lazaridou and Baroni, *Emergent Multi-Agent Communication in the Deep Learning Era*, 2020

Multi-agent (deep) reinforcement learning

Agent modelling

Agents can learn to predict the other agents' behaviour (Albrecht2018)

→ **Ficticious play** (Brown1951, Robinson1951)
- track actions of other agents to estimate their policy

→ **Bayesian learning** (Jordan1991, Foerster2019)
- track probabilities over possible policies

\+ Allows to better adapt to new partners

Albrecht and Stone, *Autonomous agents modelling other agents: A comprehensive survey and open problems*, 2018
Brown, *Iterative solution of games by fictitious play*, 1951
Robinson, *An Iterative Method of Solving a Game*, 1951
Jordan, *Bayesian learning in normal form games*, 1991
Foerster et al., *Bayesian Action Decoder for Deep Multi-Agent Reinforcement Learning*, 2019

MADRL research

- Benchmarking MADRL
- Exploration
- Zero-shot teaming
- Human-robot interaction
- Agent-based LLMs and LLM-societies

MADRL Research

Benchmarking MADRL

Hard to compare fairly all approaches.

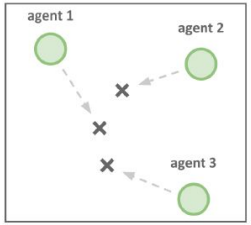
But, at the same time, research runs for best performance

- Many different environments, but not so much diversity
- Quality of envs. and tasks are not well questioned
  - What skills are we testing?
- Hard to compare with literature:
  - Many different methods, for many different settings
  - MADRL is expensive to train
  - Numerous hyperparameters (like DRL, but worse)
  - Implementation are complex, with implementation tricks (like DRL, but worse) → discrepancies in results
  - Different programming library used

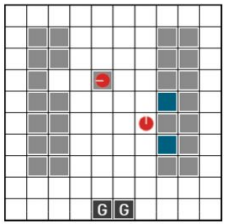
→ Attempts at benchmarking (Papoudakis2021)

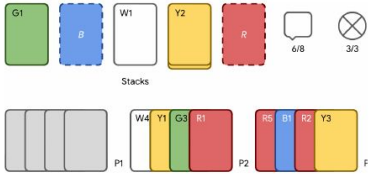
→ Attempts at instauring standardised evaluation (Gorsane2022)


(a) SMAC

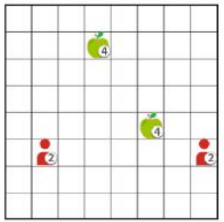


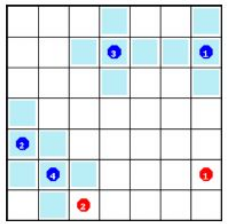
(b) MPE


(d) RWARE



(e) Hanabi


(c) LBF


(f) magym

Papoudakis et al., *Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative Tasks*, 2021
Gorsane et al., *Towards a Standardised Performance Evaluation Protocol for Cooperative MARL*, 2022

MADRL Research

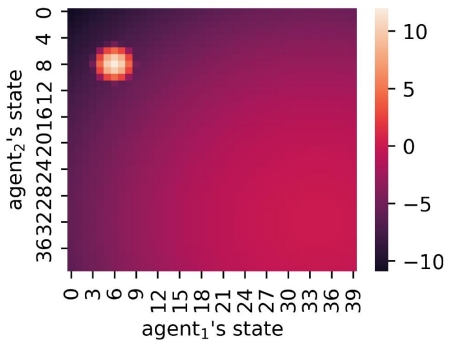
Exploration

MADRL requires exploration, like RL, but multi-agent exploration requires special care.

Problem: Exploring locally ≠ Exploring jointly

i.e., exploring the joint-observation space requires coordinated exploration

	A	B	C
A	10	−5	−5
B	−5	7	7
C	−5	7	7



Toquebiau et al., *Joint Intrinsic Motivation for Coordinated Exploration in Multi-Agent Deep Reinforcement Learning*, 2024

MADRL Research

Zero-shot teaming

Agents should be able to interact with unknown partners (Kirk2023)

Problem: Training a team of agents will results in overfitting on the team collective behaviour

→ **Agent modelling** (Albrecht2018)

→ **Population-based training** (Jaderberg2019)

→ **Predicting the "type" of observed agent** (Lanctot2023)

Albrecht and Stone, *Autonomous agents modelling other agents: A comprehensive survey and open problems*, 2018
Kirk et al., *A Survey of Zero-shot Generalisation in Deep Reinforcement Learning*, 2023
Jaderberg et al., *Human-level performance in first-person multiplayer games with population-based deep reinforcement learning*, 2019
Lanctot et al., *Population-based Evaluation in Repeated Rock-Paper-Scissors as a Benchmark for Multiagent Reinforcement Learning*, 2023

MADRL Research

Human-robot interaction

One goal of MADRL is robot application → interaction is a challenge

Understanding:

- how humans behave (Shih2021)
- how humans interact with each other (Tseng2016)
- how humans react to robots (Roesler2024)
- how humans communicate in social interactions (Feine2019)

Designing:

- physical interfaces (Zlotowski2014)
- social interfaces (Liu2022)

Shih et al., *On the Critical Role of Conventions in Adaptive Human-AI Collaboration*, 2021
Tseng et al., *Service robots: System design for tracking people through data fusion and initiating interaction with the human group by inferring social situations*, 2016
Roesler et al., *The dynamics of human–robot trust attitude and behavior — Exploring the effects of anthropomorphism and type of failure*, 2024
Feine et al., *A Taxonomy of Social Cues for Conversational Agents*, 2019
Zlotowski et al., *Anthropomorphism: Opportunities and Challenges in Human–Robot Interaction,* 2014
Liu et al., *An analysis of children' interaction with an AI chat-bot and its impact on their interest in reading*, 2022

MADRL Research

Agent-based LLMs and LLM-societies

LLMs can be prompted to follow certain behaviours.
Thus, they can be used to simulate multi-agent interactions.

- **Agent-based LLMs** (Guo2024): LLM prompted to act as an agent (human/robot/…)
  - for robotic control (Driess2023)
  - for Human-AI interaction (Liu2024)
- **LLM-society**: Groups of LLM agents
  - for simulating multi-agent interactions (Park2023, Li2023)
  - for simulating cultural evolution (Perez2024)

Guo et al., *Large Language Model based Multi-Agents: A Survey of Progress and Challenges*, 2024
Li et al., *Theory of Mind for Multi-Agent Collaboration via Large Language Models*, 2022
Liu et al., *LLM-Powered Hierarchical Language Agent for Real-time Human-AI Coordination*, 2024
Driess et al., *PaLM-E: An Embodied Multimodal Language Model*, 2023
Perez et al., *Cultural evolution in populations of Large Language Models,* 2024
Park et al., *Generative Agents: Interactive Simulacra of Human Behavior*, 2023

Thank you for you attention !

Questions ?