Distributed and Hierarchical RL

Webinar – April 24th, 2024

Gianvito Losapio, Marco Mussi, Alberto Maria Metelli, Marcello Restelli

AI4REALNET has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101119527

Outline

- Introduction (Alberto Maria Metelli)
- Distributed Reinforcement Learning (Gianvito Losapio)
- Hierarchical Reinforcement Learning (Marco Mussi)
- Research Plan (Alberto Maria Metelli)
- Q&A

Motivation

Use cases of AI4REALNET

Electricity

Railway

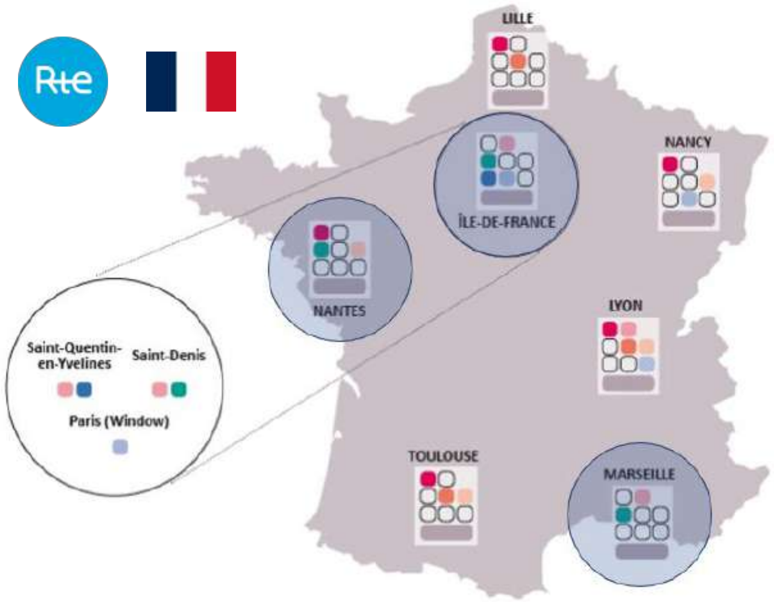
Air traffic

Two main challenges

- **Curse of dimensionality**
  Large/Infinite state and action spaces → Distributed RL
- **Curse of horizon**
  Need for planning in the far future → Hierarchical RL

Electricity and railways

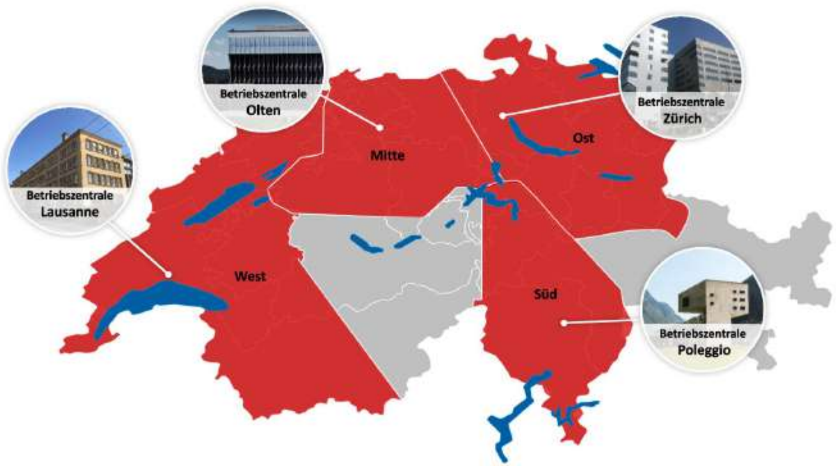
French electricity network



Control centers

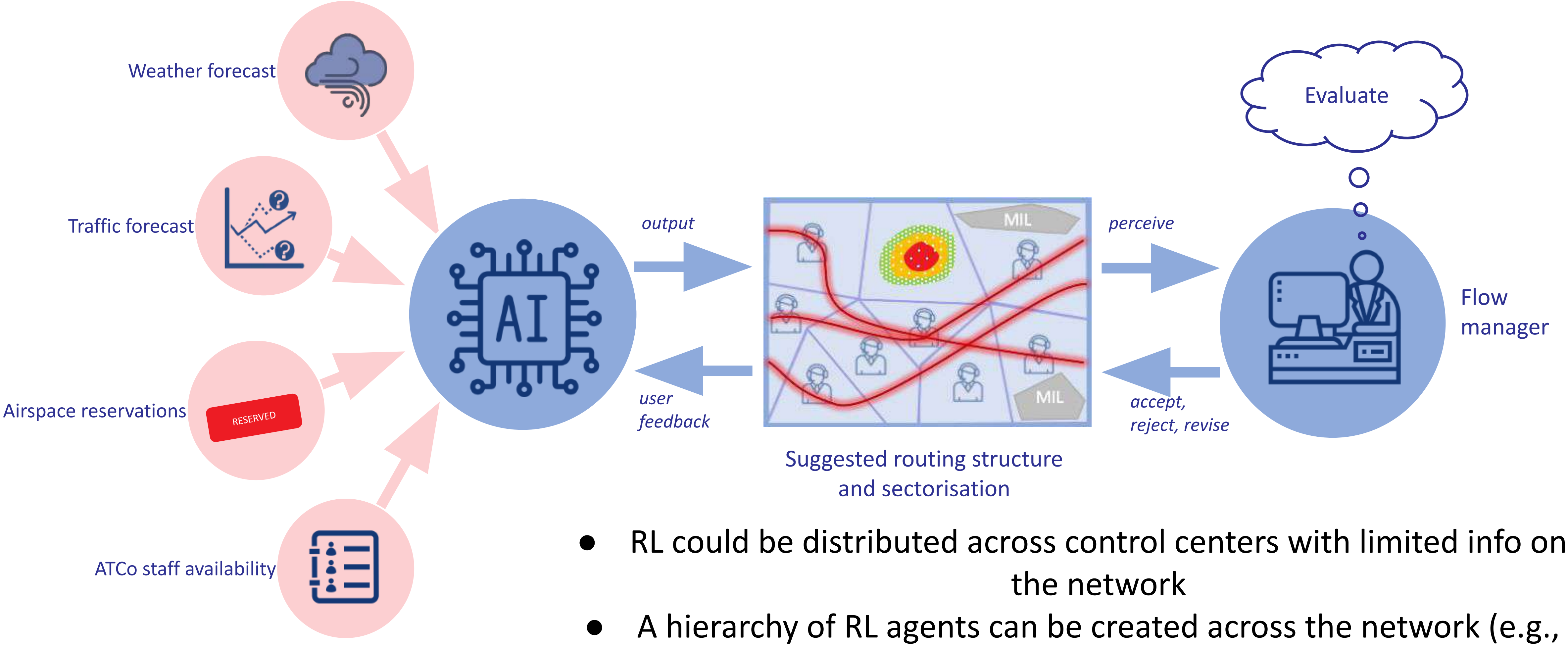
Swiss railway network

SBB CFF FFS



- RL could be distributed across control centers with limited info on the network
- A hierarchy of RL agents can be created across the network (e.g., control centers + local workers)

Air traffic



- RL could be distributed across control centers with limited info on the network
- A hierarchy of RL agents can be created across the network (e.g., control centers + local workers)

Distributed RL

Motivation

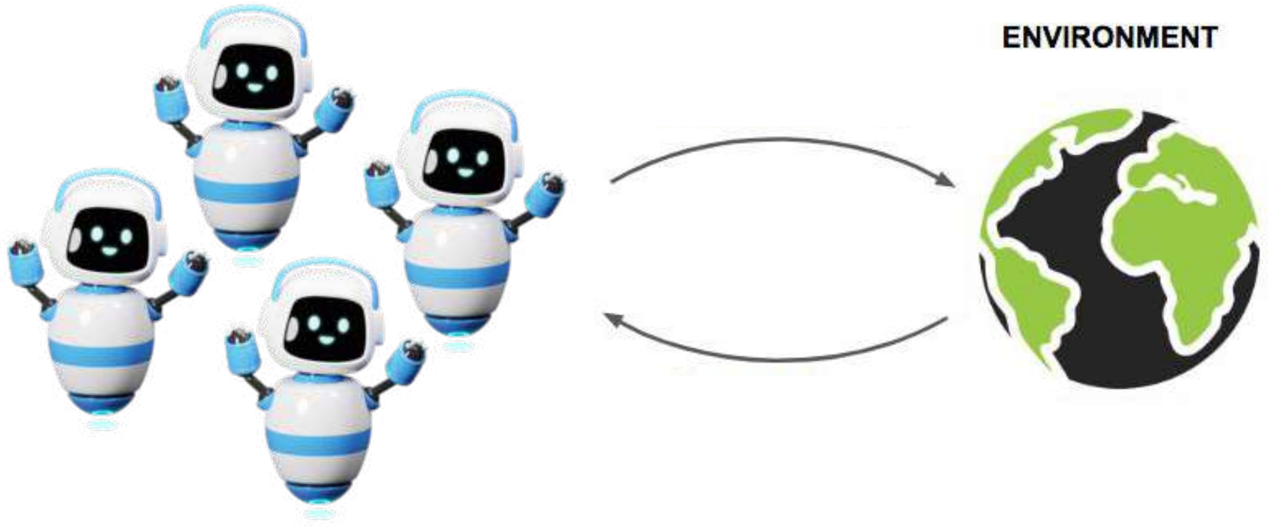
- **Curse of dimensionality**
  Large/Infinite state and action spaces ⟹ Distributed RL

- **Curse of horizon**
  Need for planning in the far future ⟹ Hierarchical RL

Definition

Distributed Reinforcement Learning (DRL) is a distributed learning process
to solve a sequential decision-making problem

Multiple agents are involved in the decision process, in such case
we refer more generally to it as **Multi-Agent Reinforcement Learning (MARL)**



Example

Robocup

https://www.robocup.org/

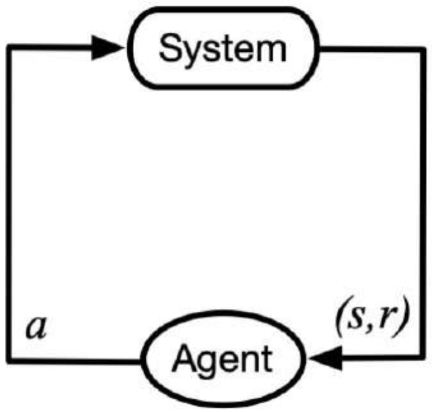
Problem formulation

Markov game

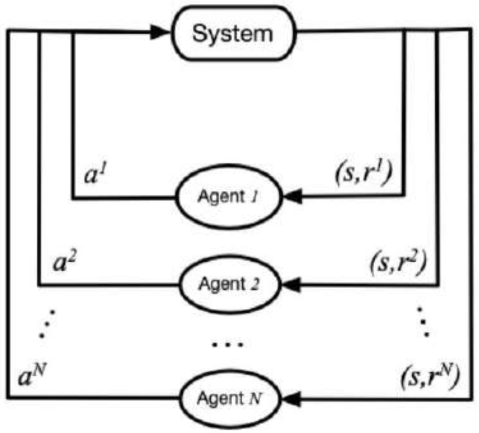
$$\mathcal{G} = \langle n, \mathcal{S}, (\mathcal{A}_i)_{i\in[n]}, P, (R_i)_{i\in[n]}, H \rangle$$

- n is the number of agents
- $\mathcal{S}$ is the set of possible states of the environment
- $\mathcal{A}_i$ is the set of possible actions available to agent i
- P is the state transition function
- R_i is the reward function of agent i
- H is the horizon

π_i is the policy of agent i



(a) Markov decision process



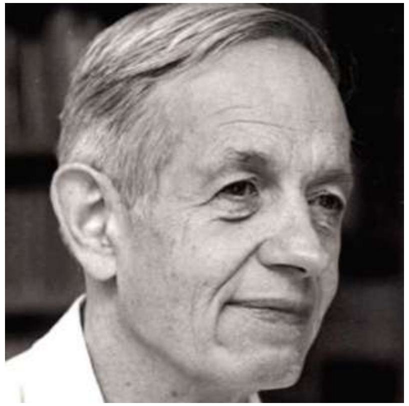
(b) Markov game

Objective

Value function of agent i

$$\underbrace{V^i_{\pi_i,\pi_{-i}}}_{\text{policies}}(s) = \mathbb{E}\left[\sum_{t=0}^{H} R_i(s_t, a_t) \,\middle|\, a_{t,i} \sim \pi_i, a_{t,-i} \sim \pi_{-i}, s_0 = s\right]$$

the performance of each agent i is controlled not only by its own policy, but also by the choices of all other agents



[Nash, 1950]

Nash equilibrium

A joint policy $\pi_* = (\pi_{1,*}, \pi_{2,*}, \ldots, \pi_{n,*})$ such that for any s, i

$$V^i_{\pi_{i,*},\pi_{-i,*}}(s) \geq V^i_{\pi_i,\pi_{-i,*}}(s) \qquad \forall \pi_i$$

MARL paradigms

There are three main settings of MARL:

- **Cooperative:** all the agents usually share and optimize the same objective

$$R_1 = R_2 = \cdots = R_n = R$$

Team game ---> need for communication

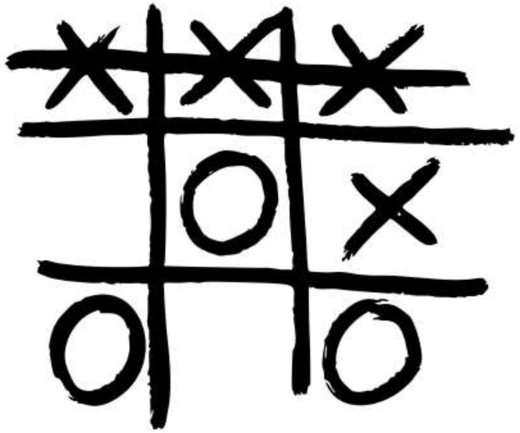
Distributed RL here

- **Competitive:** all the agents are in competition

Zero-sum Markov games

$$\sum_i R_i(s,a) = 0 \qquad \text{for any} \quad (s,a)$$

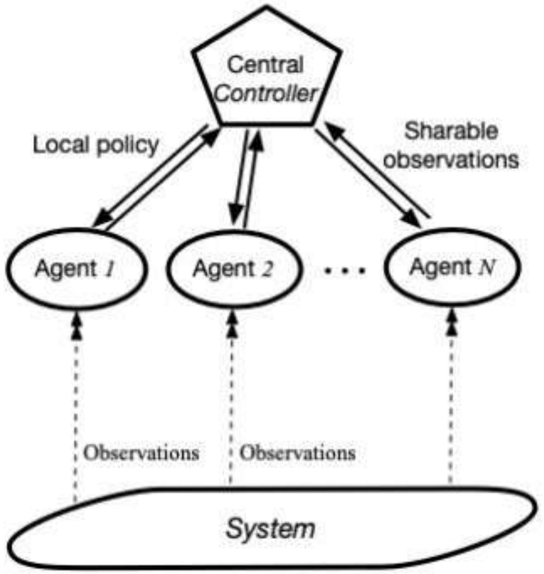
(increasing the reward of one agent makes the reward of the other agents decrease)

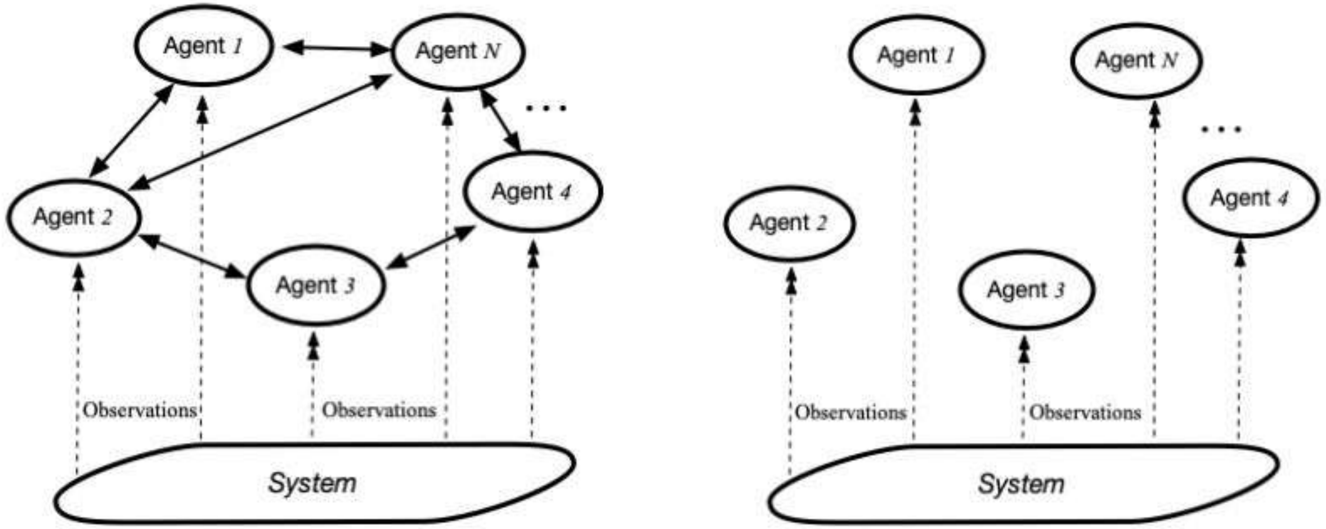


- **Mixed:** a combination of the previous two

General-sum games

How to distribute RL





Centralised training

Centralized execution

Decentralized execution

Decentralised training and execution

Communication between agents

[Stefano V. Albrecht, Filippos Christianos, Lukas Schäfer (2024). Multi-Agent Reinforcement Learning: Foundations and Modern Approaches]

Challenges of MARL

- **Non unique learning goals**
  Vague objective since NE is difficult to reach in practice

- **Non-stationarity**
  Agents usually learn concurrently

- **Multi-agent credit assignment**
  Agents contribute differently to the reward

- **Scalability**
  The joint state/action space increases exponentially with the number of agents

- **Various information structures**
  Different information available at training and execution time

Distributed Q-learning

QD-learning

Provably convergent algorithm on distributed RL with limited communication

$$Q_{t+1}^{i}(s,a) \leftarrow Q_{t}^{i}(s,a) + \alpha_{t,s,a}\left[R^{i}(s,a) + \gamma \max_{a' \in \mathcal{A}} Q_{t}^{i}(s',a') - Q_{t}^{i}(s,a)\right]$$

$$- \beta_{t,s,a} \sum_{j \in \mathcal{N}_{t}^{i}} \left[Q_{t}^{i}(s,a) - Q_{t}^{j}(s,a)\right],$$

Standard Q-learning update

Info from neighbours

[Kar, S., Moura, J. M., & Poor, H. V. (2012). Qd-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus. arXiv preprint arXiv:1205.0047.]

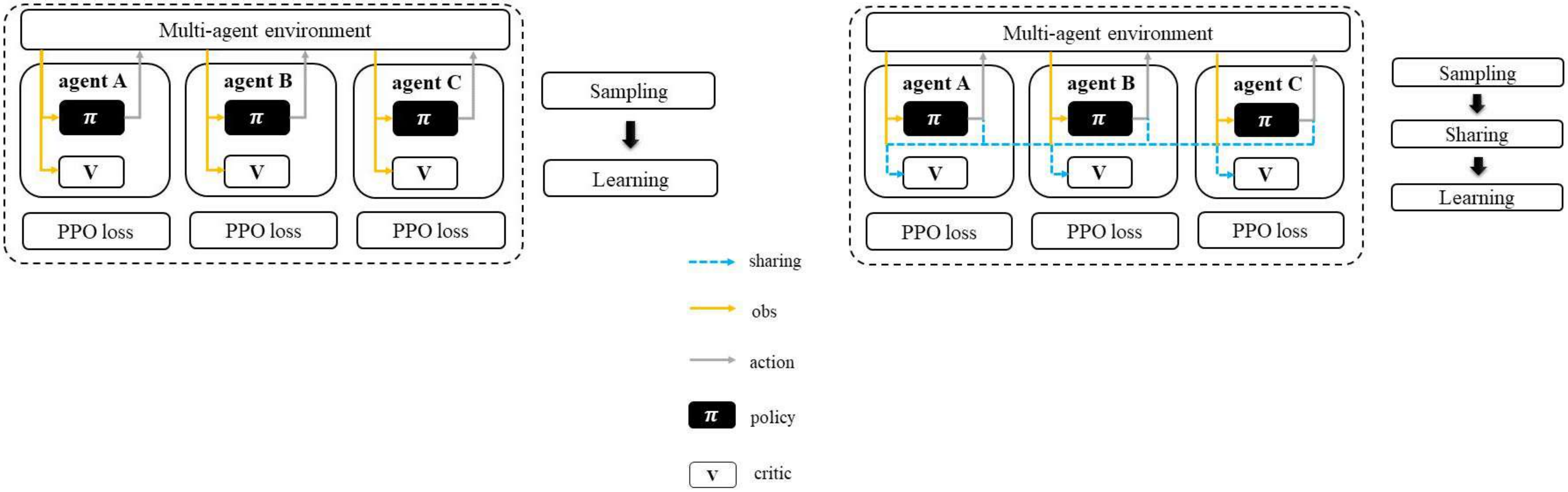
Distributing PPO

IPPO

Independent PPO agents

MAPPO

PPO with a centralized critic



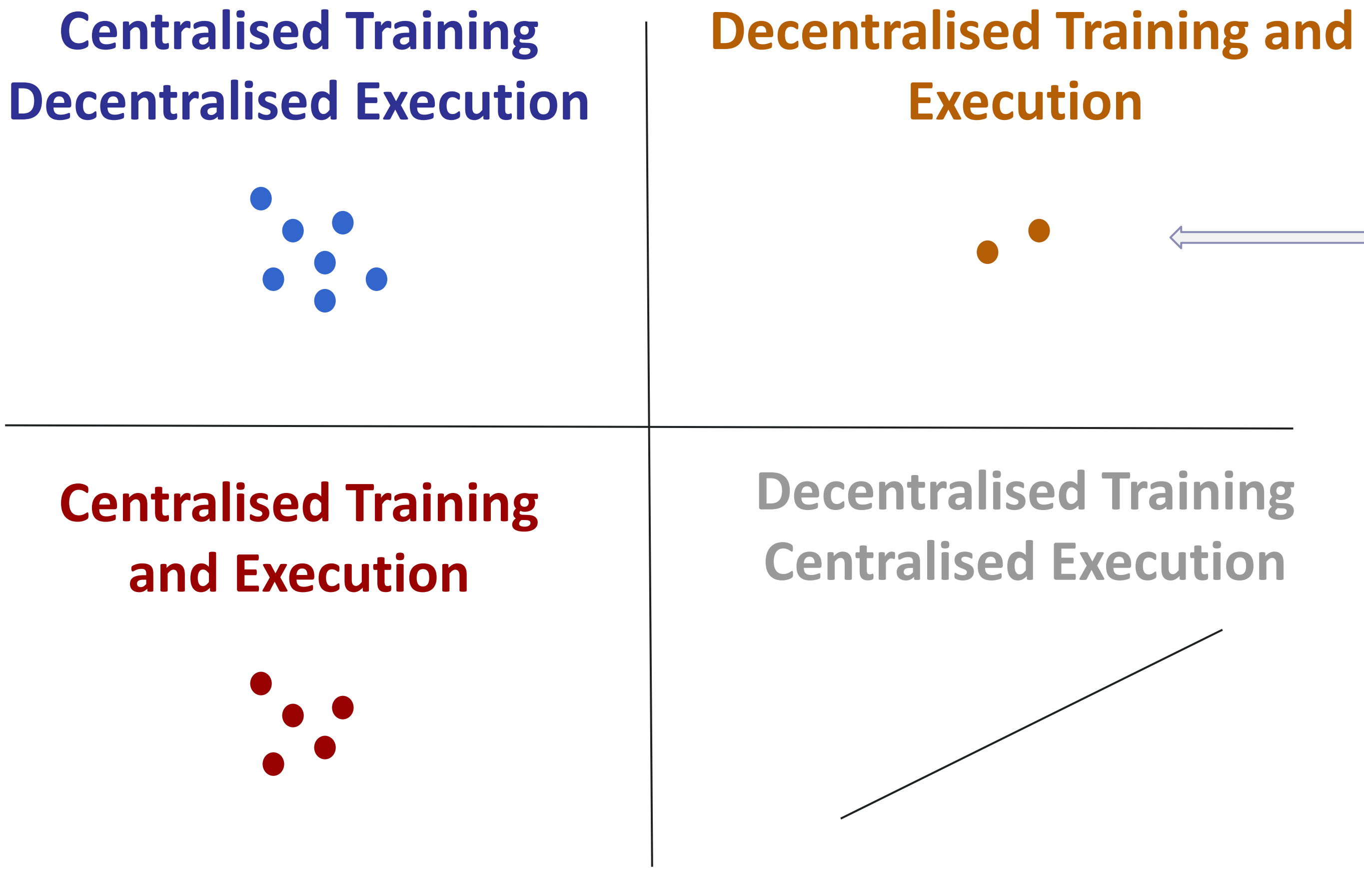
[Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., & Wu, Y. (2022). The surprising effectiveness of ppo in cooperative multi-agent games. Advances in Neural Information Processing Systems, 35, 24611-24624.]

Distributed RL - summary

- **MARL problems** can be modeled as Markov games with **Nash equilibrium** being a theoretical objective
- **Distributed RL** requires **trade-off** between independent learning and fully centralized setting
- Nowadays, almost all the solutions focus on **centralized training** and **decentralized execution**
- For the **fully-decentralized** case **PPO** gives the best results for independent learning in non-stationary environments
- One of the most important questions is about how to **communicate** in MARL

Distributed RL - map

Number of state-of-the-art algorithms per category



Only few algorithms!

[Stefano V. Albrecht, Filippos Christianos, Lukas Schäfer (2024). Multi-Agent Reinforcement Learning: Foundations and Modern Approaches]

Future research directions

- **Multimodal communication**
  Heterogeneous source of information
- **Model-based algorithms**
  Very few algorithms exist in literature
- **Inverse RL for distributed problems**
  Understanding rewards
- **Safe algorithms**
  Imposing safety constraints on training/execution
- **Usage of Large Language Models (LLMs)**
  Using recent prompting methods to generate actions

Hierarchical RL

From Large state spaces to long horizons

- **Curse of dimensionality**
  Large/Infinite state and action spaces ⟹ Distributed RL

- **Curse of horizon**
  Need for planning in the far future ⟹ Hierarchical RL

Challenges

Challenges:

- **Exploration** over large horizons
- **Credit assignment** over large horizons

Solution:

- Create a **hierarchical control structure**
- Reduce the **long-horizon** problem into a sequence of **short-horizon** ones

Example - Going on Holidays

High-Level Goals

Low-Level Tasks

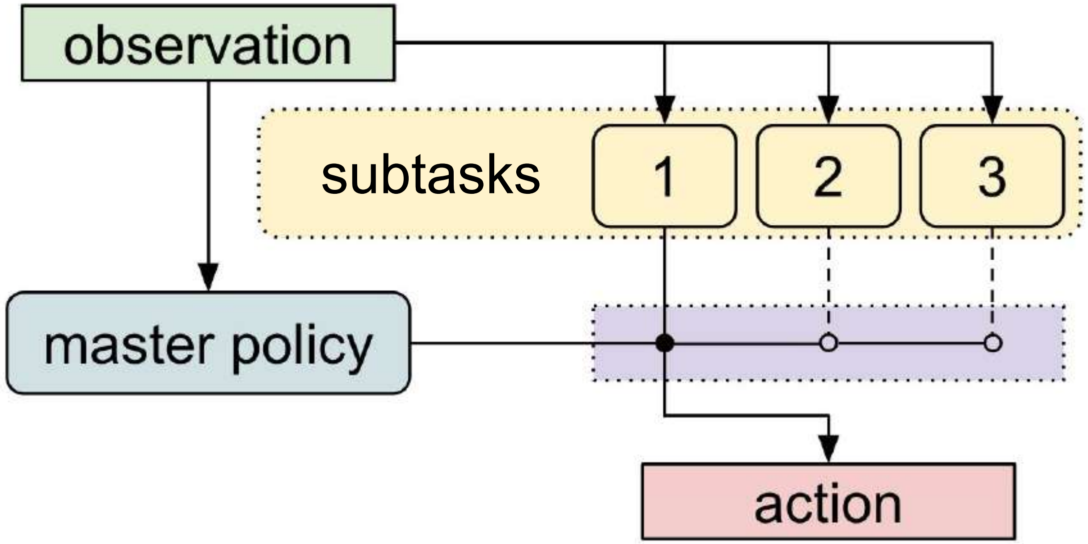
Book Tickets
- Open Booking Website
- Enter Flight Information
- …

Go to the Airport
- Go to Taxi Stand
- Call a Taxi
- …

Pateria, S., Subagdja, B., Tan, A. H., & Quek, C. (2021). Hierarchical reinforcement learning: A comprehensive survey. ACM Computing Surveys, 54(5), 1-35.

Definition

Hierarchical Reinforcement Learning (HRL) is learning to solve long-term sequential decision-making problems by decomposing them into a hierarchy of simpler subtasks



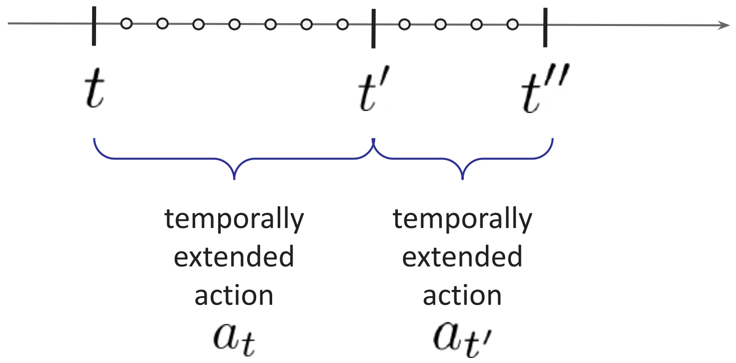
Pateria, S., Subagdja, B., Tan, A. H., & Quek, C. (2021). Hierarchical reinforcement learning: A comprehensive survey. ACM Computing Surveys, 54(5), 1-35.

Problem formulation

Semi-Markov Decision Process (SMDP)

$$\boxed{\mathcal{M}_S = \langle \mathcal{S}, \bar{\mathcal{A}}, \bar{P}, \bar{R}, H \rangle}$$

- $\mathcal{S}$ is the set of possible states of the environment
- $\bar{\mathcal{A}}$ is the **temporally extended** action space (a.k.a. **subtasks**)
- $\bar{P}$ is the state transition function (state reached after playing the temporally extended action and next time step t')
- $\bar{R}$ is the reward function accumulated until the end of the temporally extended action
- H is the horizon



Drappo, G., Metelli, A. M. & Restelli, M. (2023). An Option-Dependent Analysis of Regret Minimization Algorithms in Finite-Horizon Semi-MDP. *Transactions on Machine Learning Research*, *1*, 1-1.

Problem formulation

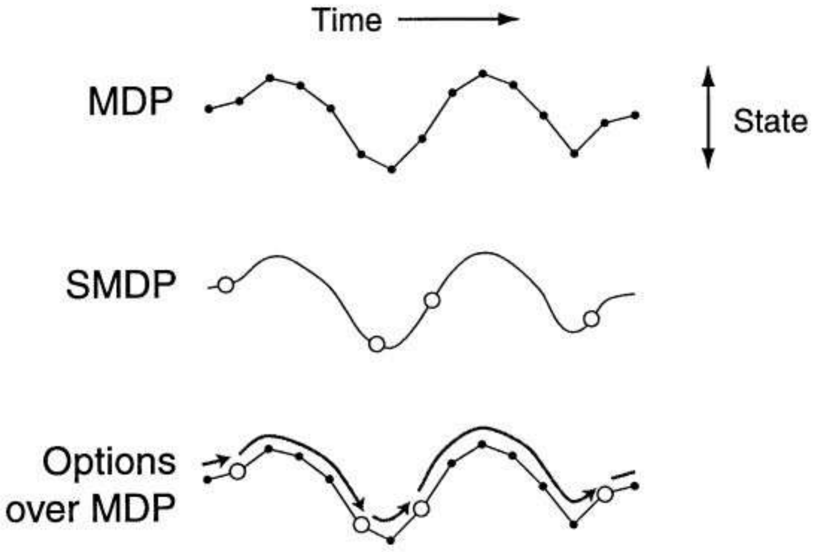
Options

$$o = \left(\mathcal{I}_o, \beta_o, \pi_o\right)$$

A possible formalization of a temporally extended action

- Option activates in certain states selected by **high-level policy** → initiation condition $\mathcal{I}_o$
- Plays an inner **low-level policy** π_o
- **Termination** condition β_o

- Each option solves a "**subtask**" (may itself be a classical RL problem)



Sutton, R. S., Precup, D., & Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, *112*(1-2), 181-211.

Objective

$$\Omega_h^*, \pi_h^* \in \arg\max_{\Omega} \max_{\pi \mid \Omega} \mathbb{E}_{a_t \sim \pi \mid \Omega} \left[\sum_{t=0}^{H} R(s_t, a_t) \right]$$

subtask space

hierarchical policy

1. Learning a hierarchical policy given a subtask space
(*state-to-subtask-to-action mapping*)

2. Subtask discovery
(*learning the optimal subtask space*)

Subtask discovery

- We can decide whether to learn or not the subtask
    - Subtask can also be hand-crafted
        - Learning the optimal policy can be difficult also in this basic case due to the challenges in: reward propagation, value function decomposition, state/action space design
    - In order to reach full automation, we aim at learn optimal subtasks

Learning hierarchical policy

Feudal hierarchy

High-level policy

⇩

subgoal
corresponding to any subtask

⇩

Worker

⇩

universal policy
learn all the possible subtasks
by taking the corresponding subgoals

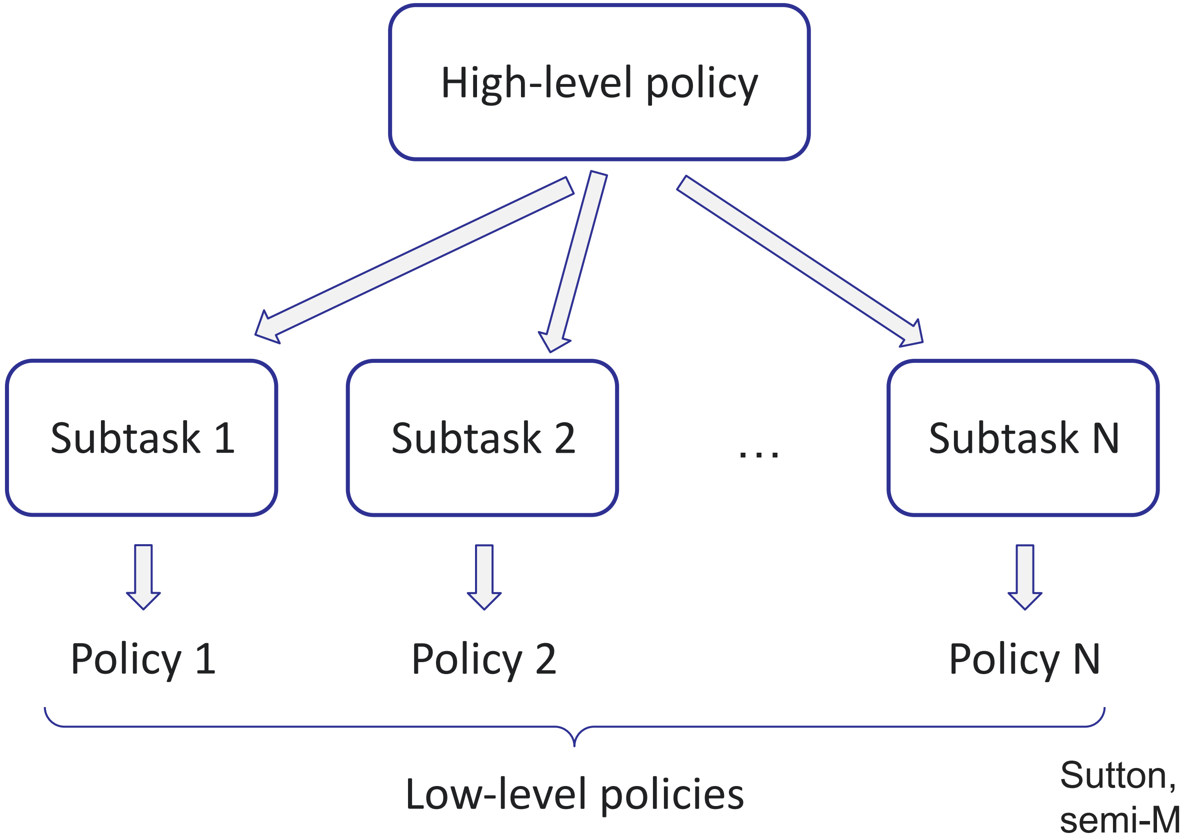
- The action space of the high-level policy consists of **subgoals** corresponding to various **subtasks**

- A **subgoal** chosen by the high-level policy is taken as input by a **universal policy** at lower level

Dayan, P., & Hinton, G. E. (1992). Feudal reinforcement learning. *Advances in neural information processing systems*, *5*.

Learning hierarchical policy

Policy tree



- The action space of the high-level policy consists of the **different low-level policies** of the **subtask**
- A **subtask** in this case has **its own policy**

Sutton, R. S., Precup, D., & Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, *112*(1-2), 181-211.

Multi-agent HRL

- **Idea**: split a joint task into **multiple subtasks** distributed across **different HRL agents**
  - HRL agents **learn to coordinate** their **high-level policies**

- Same paradigms of standard MARL
  - **Centralized/decentralized training** and **decentralized execution**

- Additional challenges:
  - **Synchronization** of **subtask terminations** across different agents
  - **Subtask space** may become **non-stationary** due to other agents

Challenges of HRL

- **Learning at various levels**
  Reward propagation, value function decomposition, state/action space design

- **Non-stationarity**
  Simultaneously changing policies at different levels

- **Global optimality**
  Ensuring the optimality of the hierarchical policy as a whole

- **Learning various components of subtasks**
  Termination/initiation conditions, subgoals

- **Theoretical support**
  Understand advantage of HRL in terms of optimal performance

Pateria, S., Subagdja, B., Tan, A. H., & Quek, C. (2021). Hierarchical reinforcement learning: A comprehensive survey. ACM Computing Surveys, 54(5), 1-35.

Hierarchical RL - summary

- **HRL problems** can be modeled as **Semi-Markov Decision Processes**, with **options** being one possible formalization

- HRL consists of **two sub-problems**
  - Learning a **hierarchical policy** given a subtask space
  - **Subtask space** discovery

Conclusion

Research plan

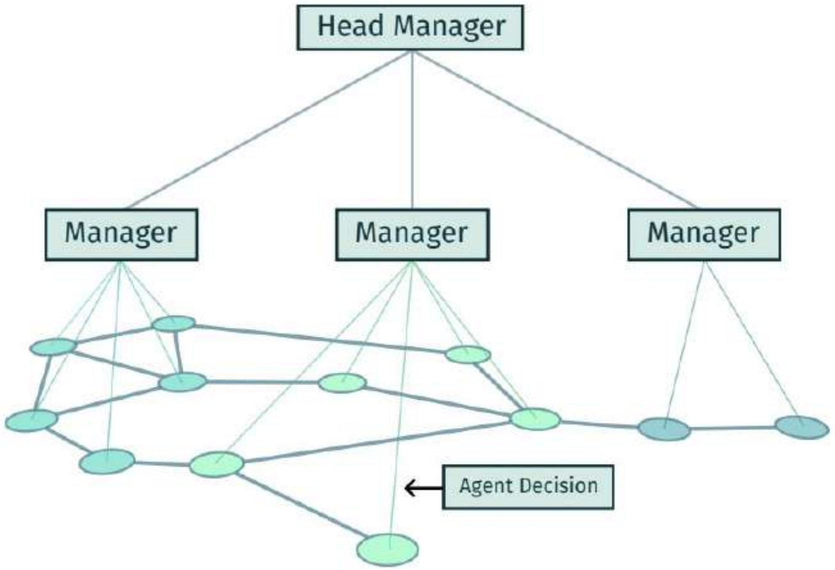
Most promising research directions for AI4REALNET

- Identify in a *data-driven way the decentralized decomposition* of the problem that minimizes the introduced bias

- Extension of the state-of-the-art algorithms to decentralized approach with *limited communication*

- Extension of the state-of-the-art *temporal abstraction* approaches to the *policy search* class of reinforcement learning algorithms

- Identify the *minimum amount of information* to be shared among agents in order to induce the desired behavior



References

References

- Stefano V. Albrecht, Filippos Christianos, Lukas Schäfer (2024). Multi-Agent Reinforcement Learning: Foundations and Modern Approaches

- Zhang, K., Yang, Z., & Başar, T. (2021). Multi-agent reinforcement learning: A selective overview of theories and algorithms. Handbook of reinforcement learning and control, 321-384.

- Kar, S., Moura, J.M., Poor, H.V.: QD-learning: A collaborative distributed strategy for multiagent reinforcement learning through consensus + innovations. IEEE Transactions on Signal Processing 61(7), 1848–1862 (2013)

- Changxi Zhu, Mehdi Dastani, Shihan Wang (2024). “A Survey of Multi-Agent Deep Reinforcement Learning with Communication.” In: Autonomous Agents and Multi-Agent Systems, vol. 38, no. 4.

- Afshin Oroojlooy, Davood Hajinezhad (2023). “A Review of Cooperative MultiAgent Deep Reinforcement Learning.” In: Applied Intelligence , vol. 53, pp. 13677-13722.

- Annie Wong, Thomas Bäck, Anna V. Kononova, Aske Plaat (2023). “Deep Multiagent Reinforcement Learning: Challenges and Directions.” In: Artificial Intelligence Review , vol. 56, pp. 5023-5056.

- Hu, S., Zhong, Y., Gao, M., Wang, W., Dong, H., Liang, X., ... & Yang, Y. (2023). Marllib: A scalable and efficient multi-agent reinforcement learning library. Journal of Machine Learning Research, 24(315), 1-23.

- Pateria, S., Subagdja, B., Tan, A. H., & Quek, C. (2021). Hierarchical reinforcement learning: A comprehensive survey. ACM Computing Surveys (CSUR), 54(5), 1-35.

AI4REALNET has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101119527