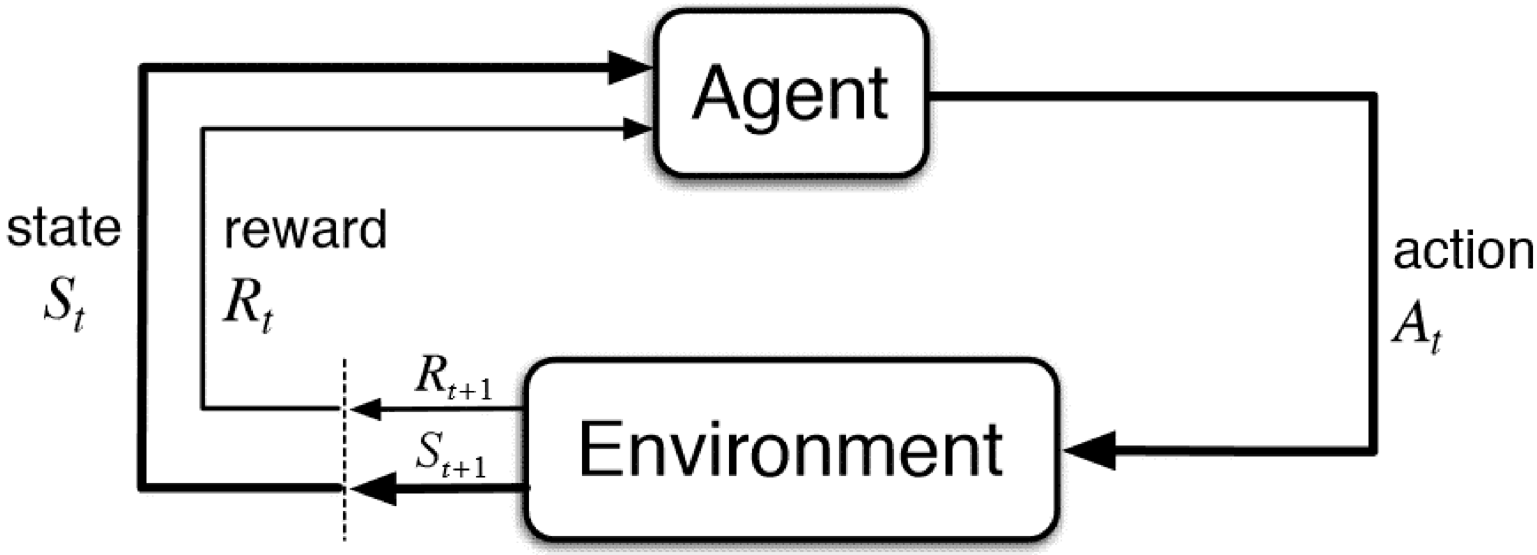
Action Trading for Self-Interested Multi-Agent Reinforcement Learning in an Escape Room Setting

Arnold Unterauer



Source: Richard S. Sutton and Andrew G. Barto **Reinforcement Learning: An Introduction**

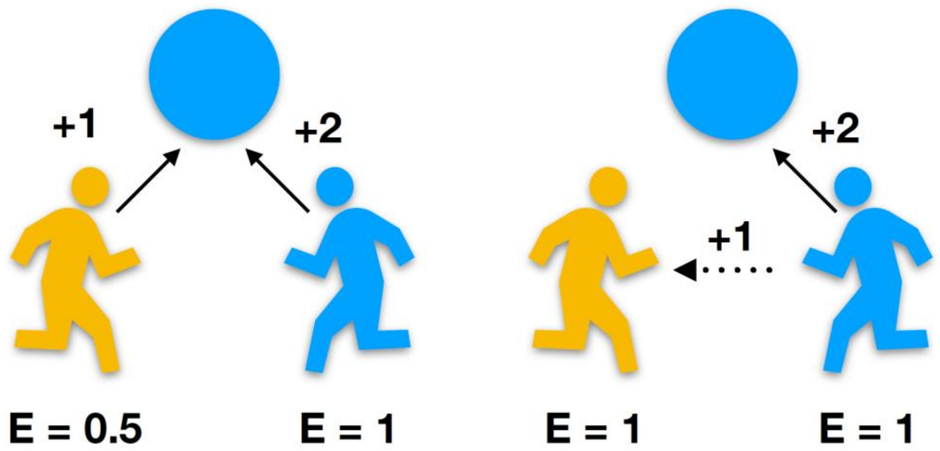
Multi Agent Systems:

agents maximize their own reward
selfish bahaviour
no cooperation between agents
unused potential

Solution:

Cooperative Game Theory
enable cooperation between agents
motivate agents to cooperate

Related Work – Action Markets



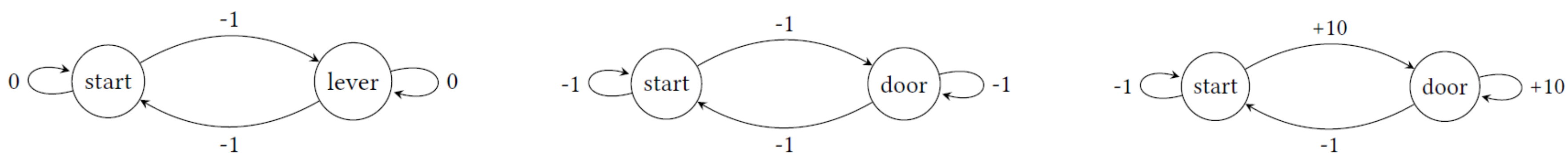
agents trade reward for the following action

agents are able to trade with extended action space

cooperating agents outperformed selfish ones

Source: Kyrill Schmid, Lenz Belzner, Thomas Gabor, and Thomy Phan **Action Markets in Deep Multi-Agent Reinforcement Learning**

Related Work – Learning to Incentivize Other Learning Agents



(a) Agent A2 is penalized for any change of state, if not receiving reward from A1.

(b) Agent A1 is penalized at every step if A2 does not pull the lever.

(c) A1 get +10 and terminates the episode by going to the door if A2 pulls the lever.

agents have to escape from a room by opening the door

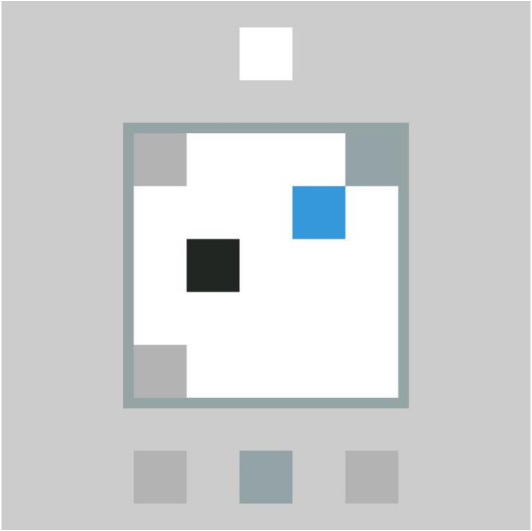
the lever for the door can only be pulled by one agent, which results in a penality

the other agent is punished for every step inside the room

→ agent will not pull the lever, everyone is trapped

→ cooperation is needed to overcome this hurdle

Source: Jiachen Yang, Ang Li, Mehrdad Farajtabar, Peter Sunehag, Edward Hughes and Hongyuan Zha **Learning to Incentivize Other Learning Agents**



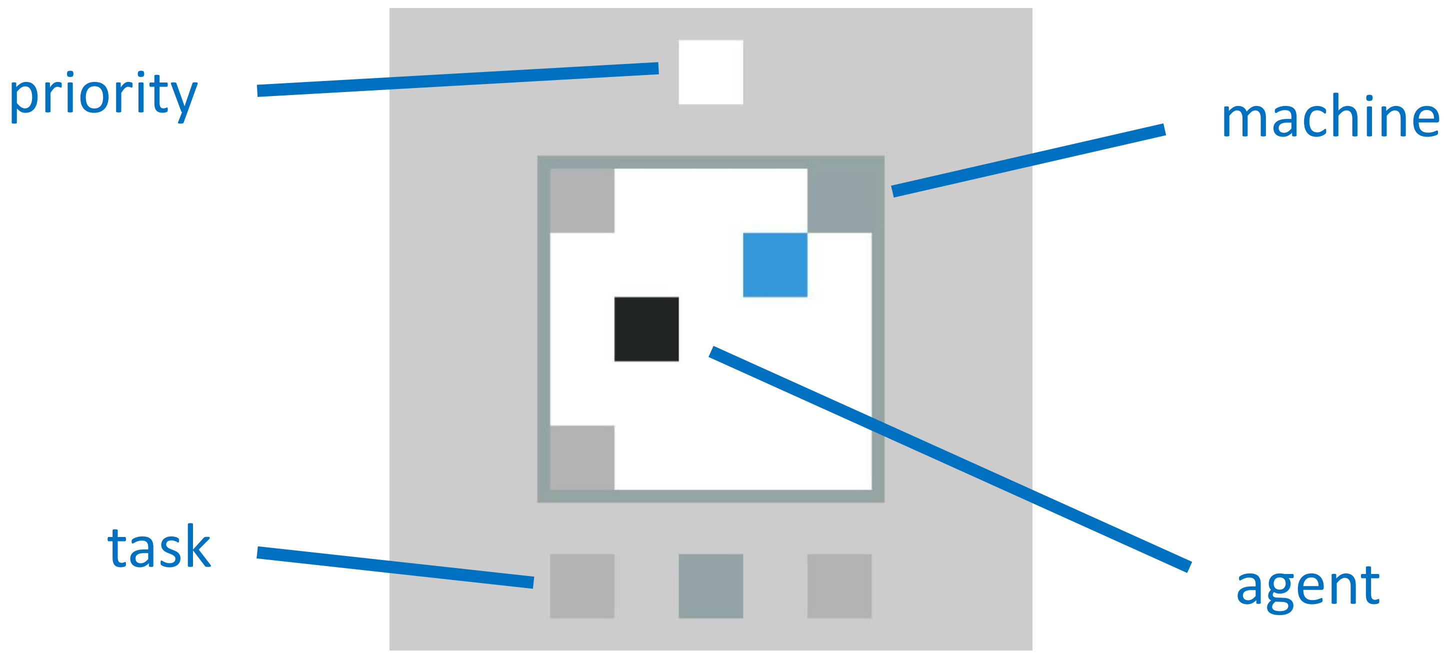
Smart Factory:

Complete tasks by processing machines fast

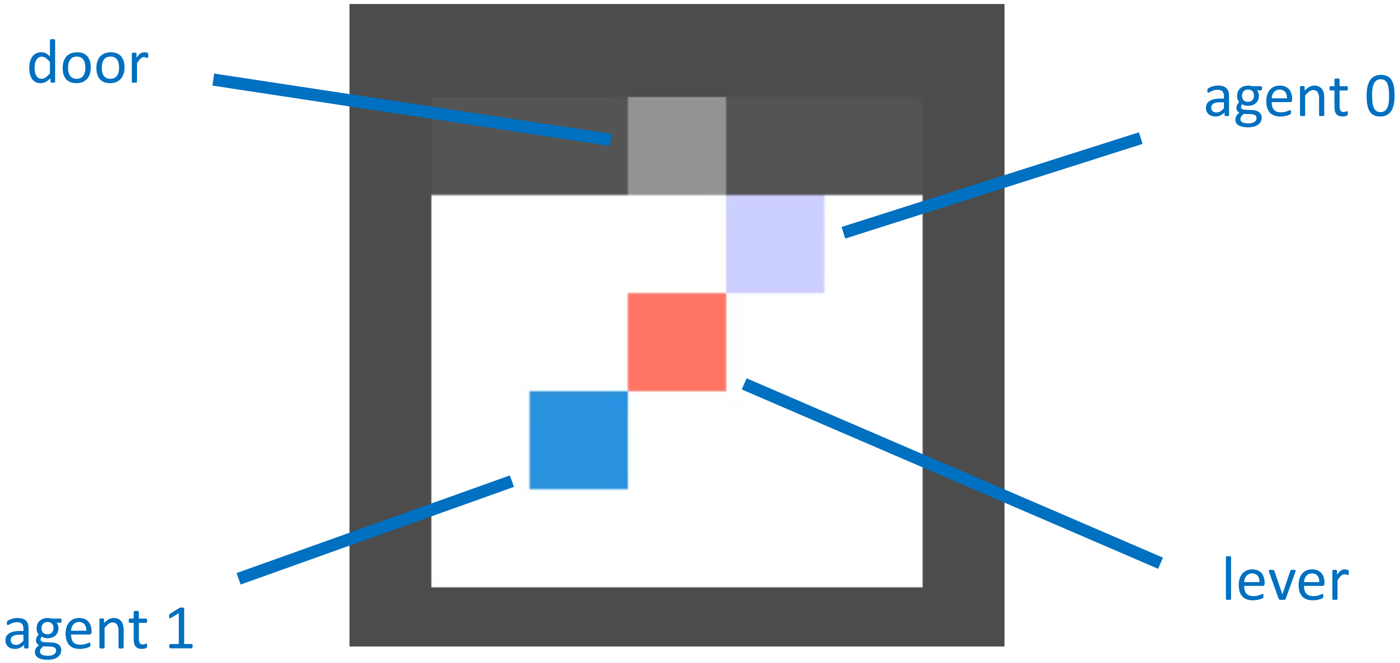
Escape Room:

Leave the room by cooperating to pull the lever

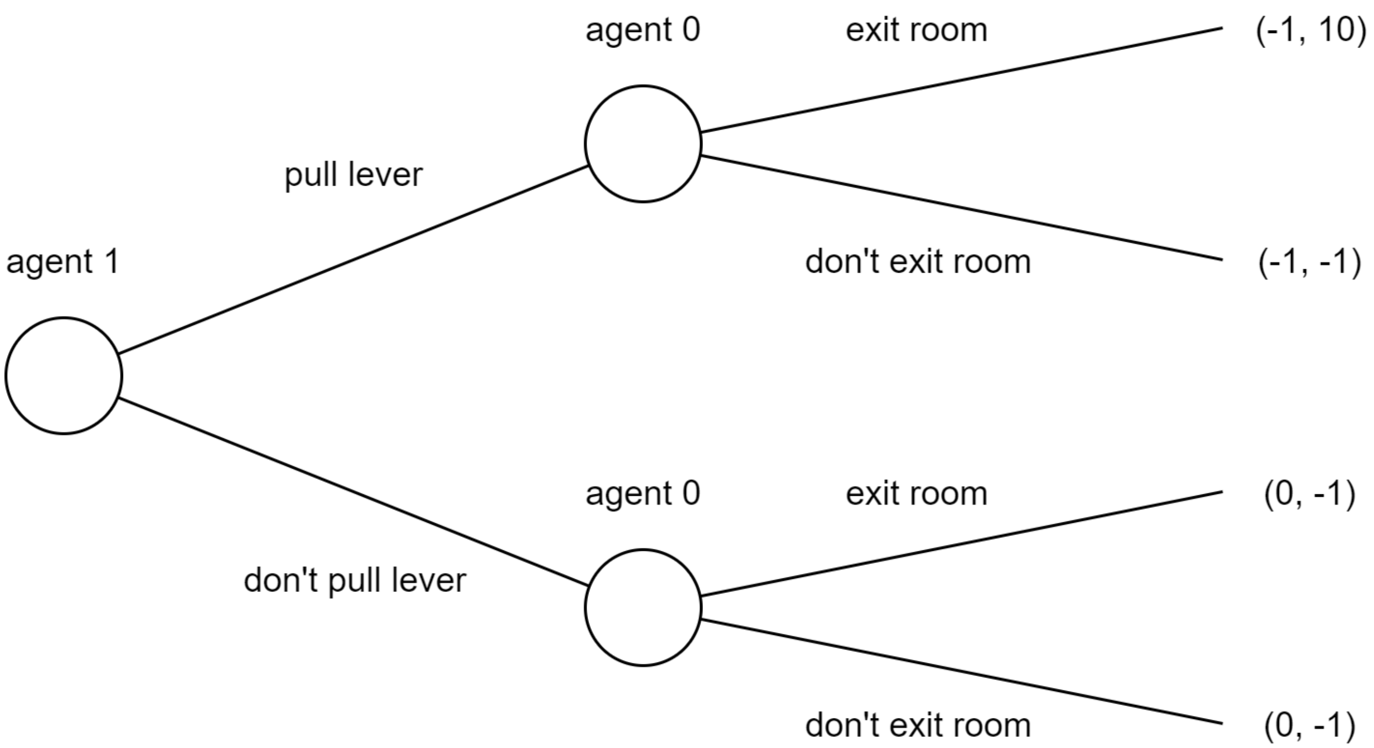
Environment – Smart Factory



Environment – Escape Room



Environment – Escape Room



Environment – Escape Room

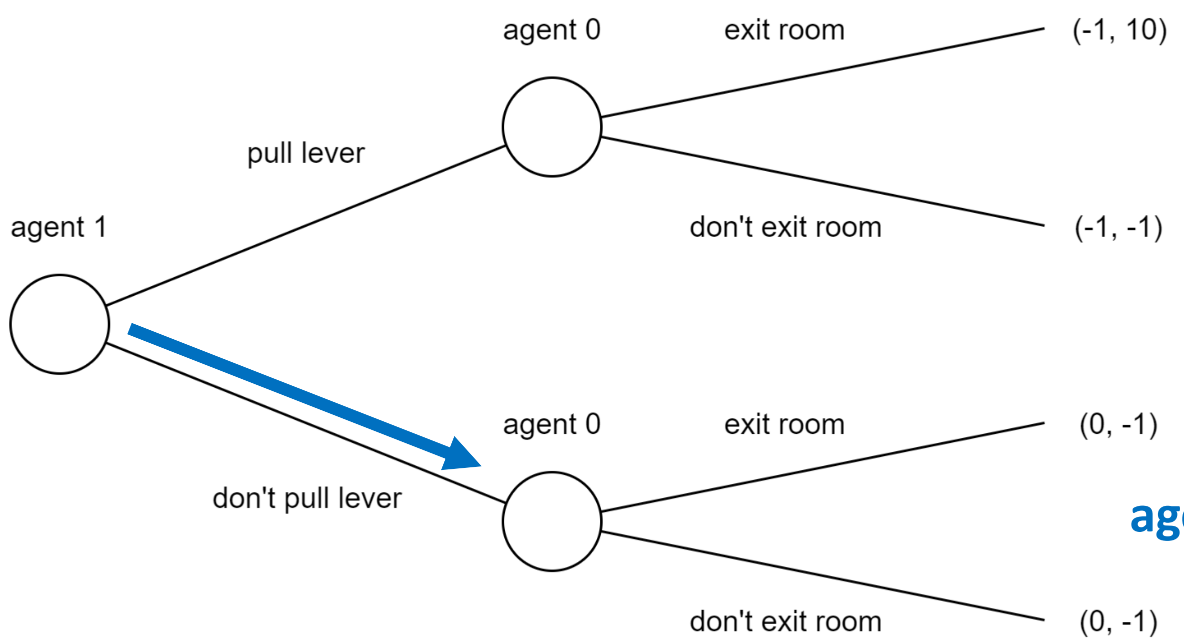
agent 1 \ agent 0	exit	don't exit
pull	-1 / 10	-1 / -1
don't pull	0 / -1	0 / -1

Expected rewards:

agent 1	pull: -1	don‘t pull: 0
agent 0	exit: 4.5	don‘t exit: -1

Environment – Escape Room

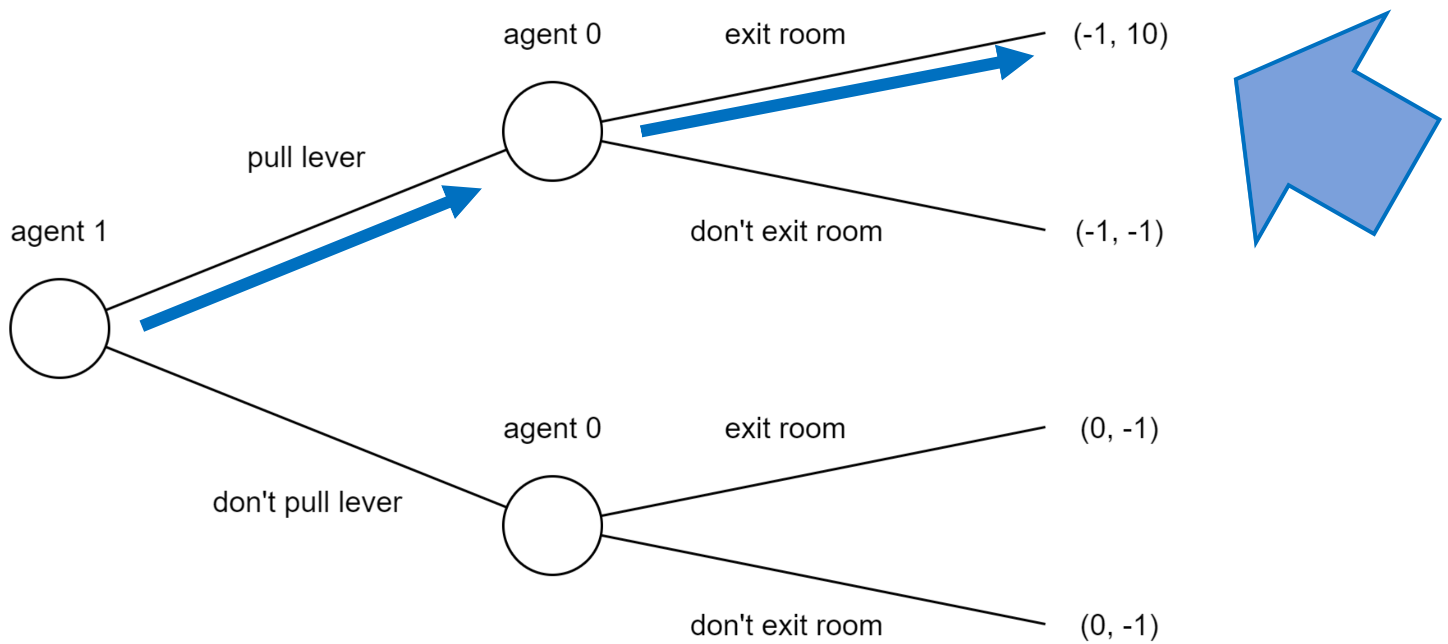
agent 1	pull: -1	don‘t pull: 0
agent 0	exit: 4.5	don‘t exit: -1



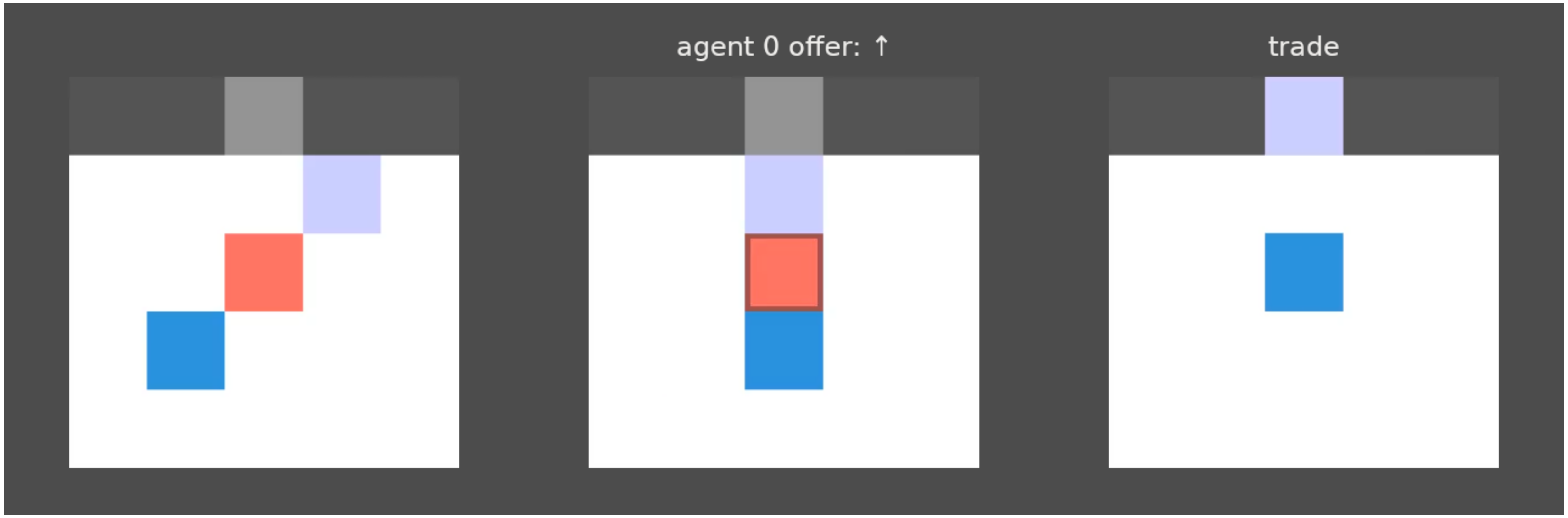
agent 0 reward: -1

Environment – Escape Room

agent 1	pull: -1 + C	don‘t pull: 0
agent 0	exit: 4.5	don‘t exit: -1



Action Trading



agent makes an offer to other agent

other agent can perform the offer action

by executing the agent gets compensated

Compensation

$$Compensation = M_c * \frac{Q_{max} - Q_{offer}}{\gamma}$$

Valuation Networks: pretrained non-cooperating agent networks

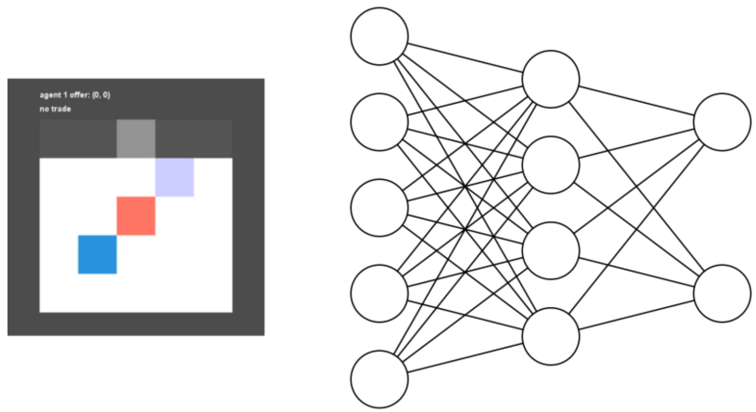
Policy Networks: learning agents

Target Networks: learning agents

Fixed Value: fixed compensation amount

Experiments

Deep Q-Network



action

128 64 32

Decaying ε-greedy: over 6000 epsiodes to ε_{min} = 0.01

10 runs: 6000 episodes with γ = 0.95

Fixed Compensation: 2

Mark up: $M_c = 1.1$

Smart Factory:

- random selected priorities

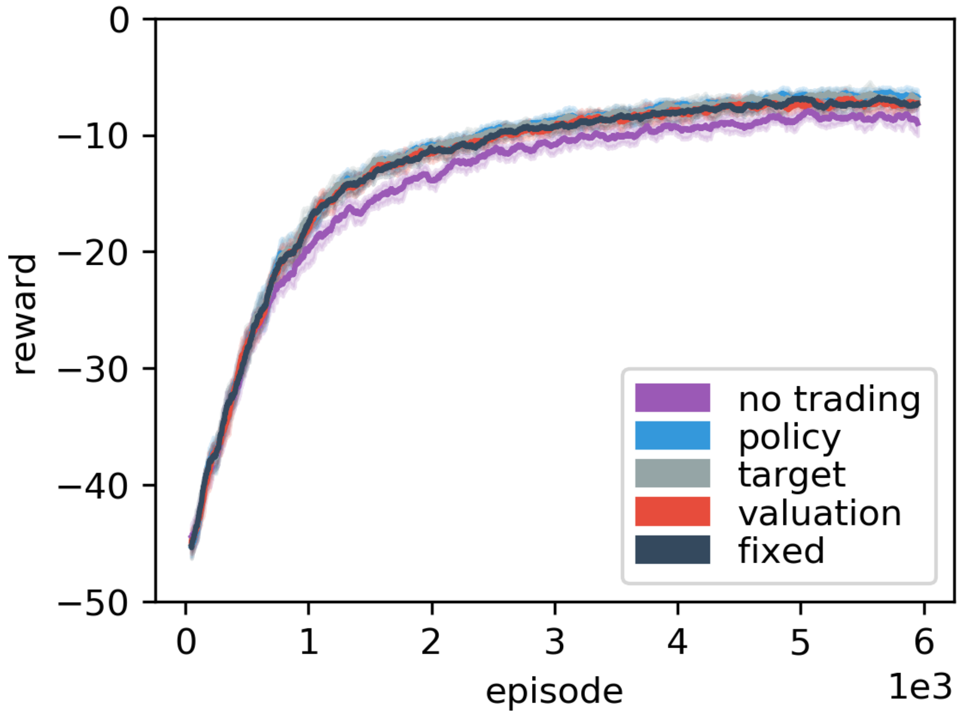
	high priority	low priority
step in factory	-0.5	-0.02
complete task	1	1

Escape Room:

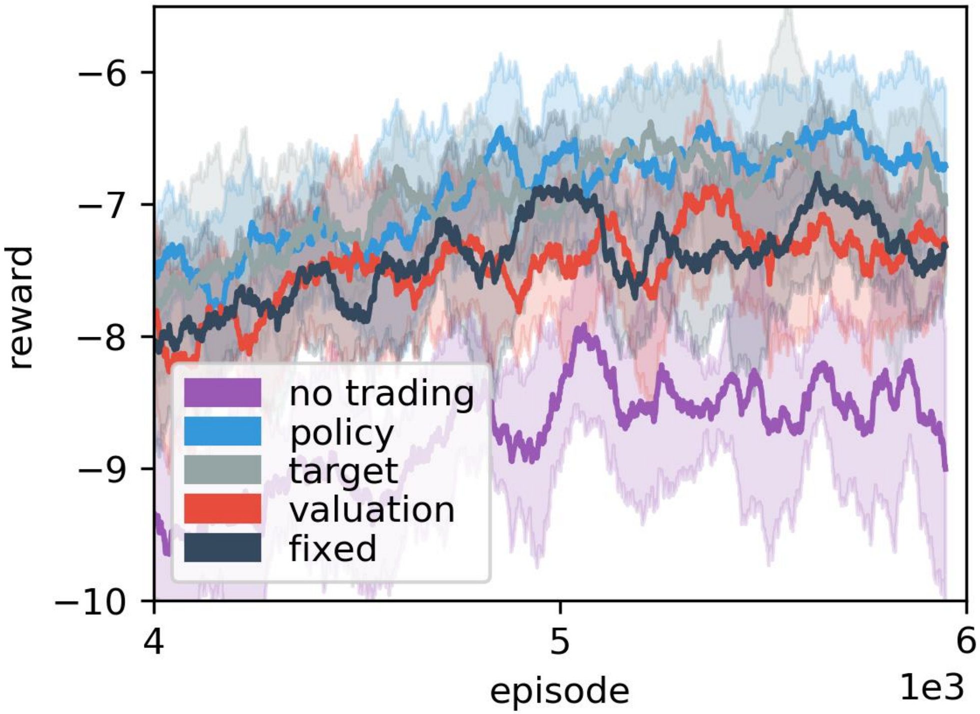
- fixed wall direction
- random generated wall direction

	agent 0	agent 1
step in room	-1	0
pull lever	0	-1
exit room	10	0

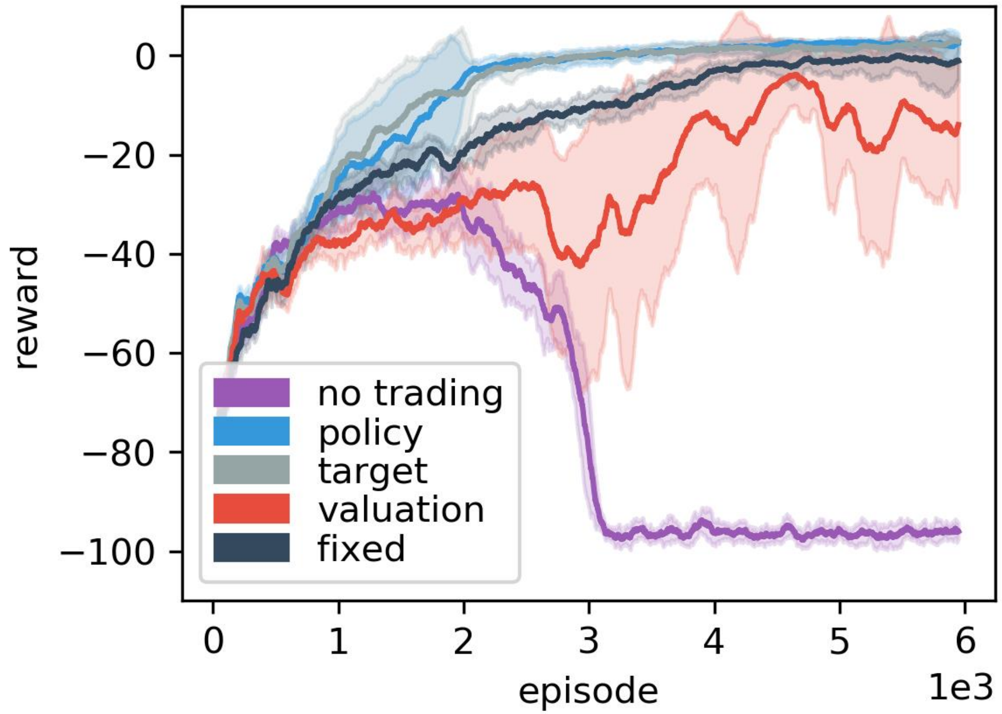
Smart Factory Training



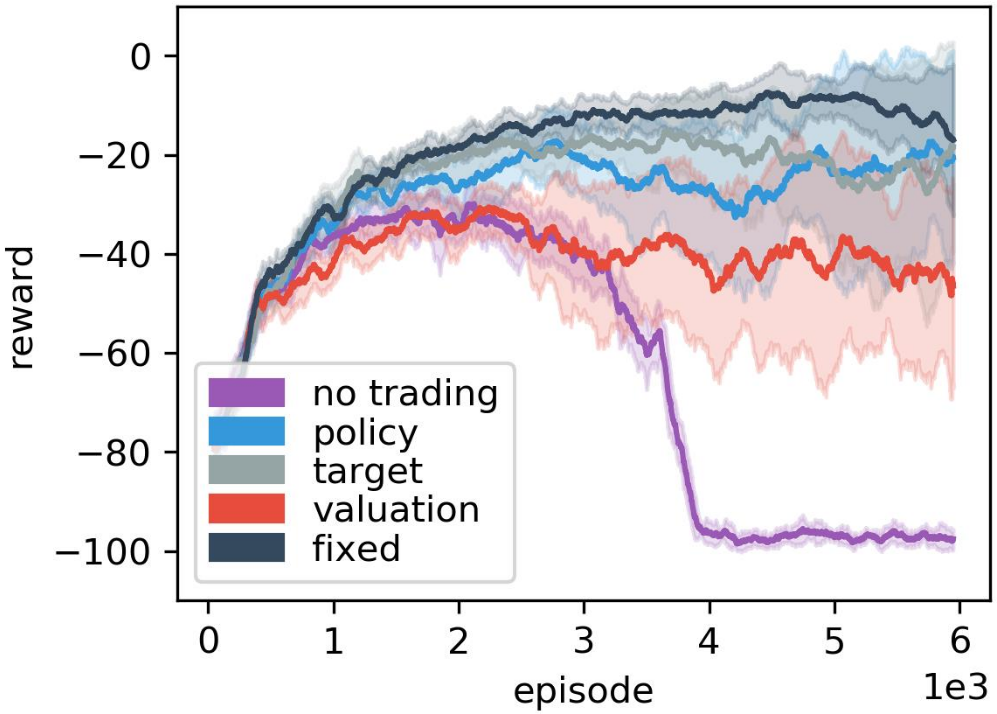
Smart Factory Training Close-up



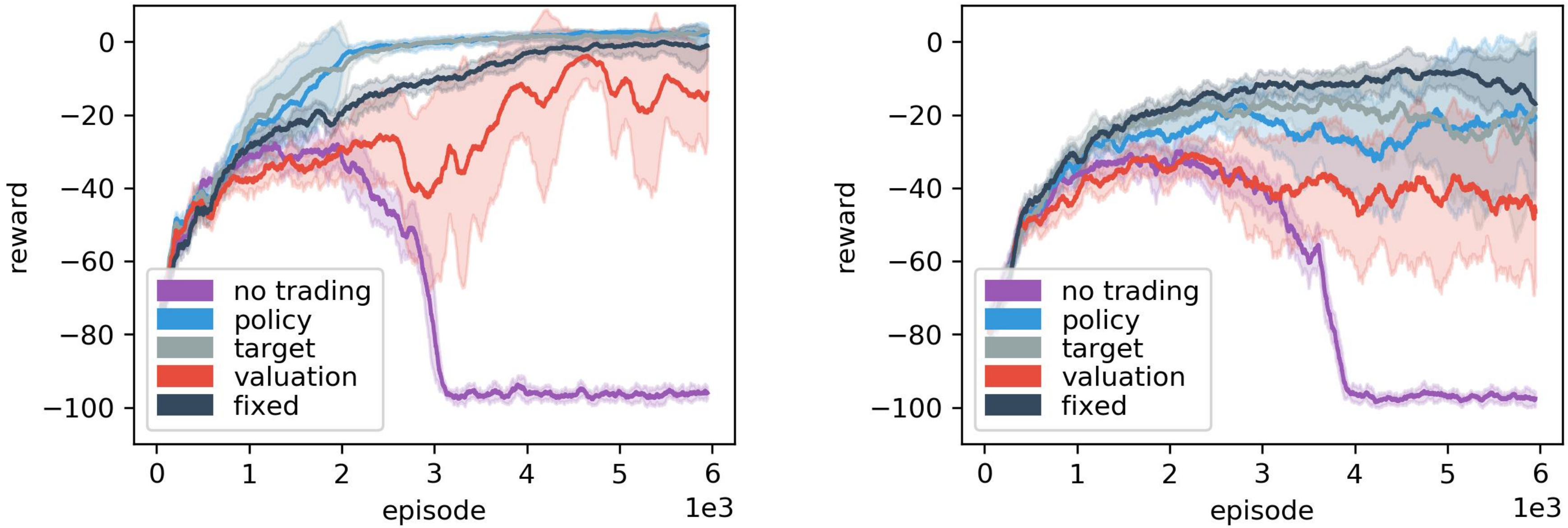
Escape Room fixed wall direction



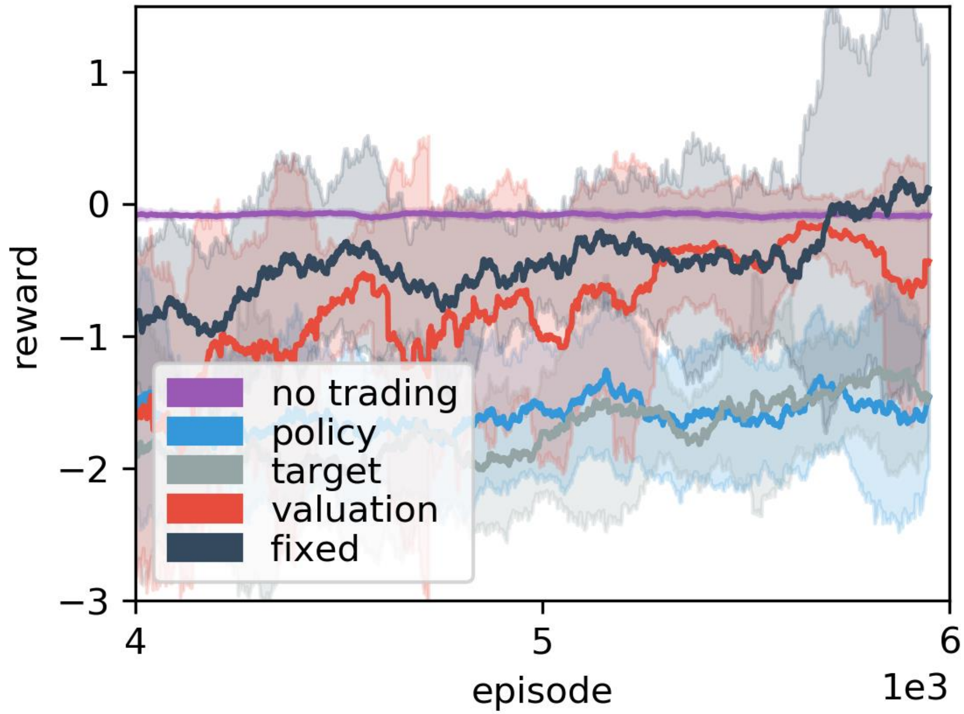
Escape Room random wall direction



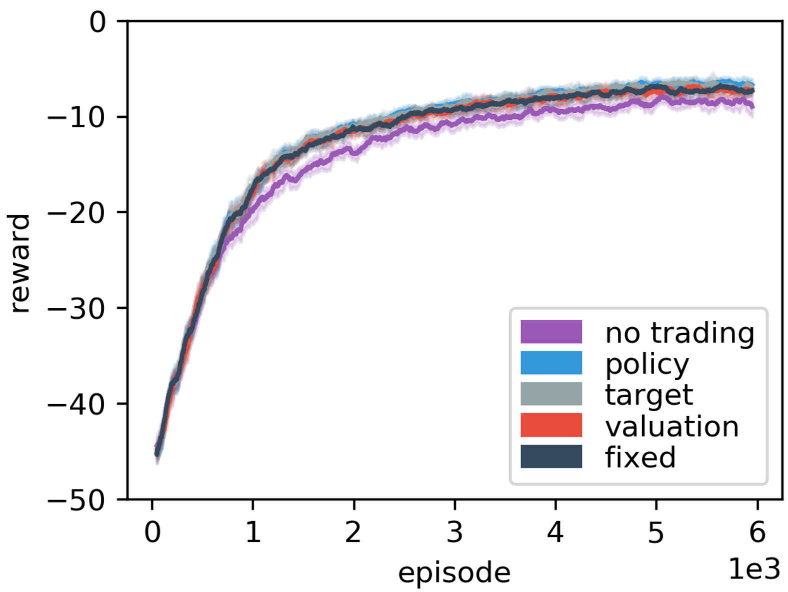
Escape Room comparison

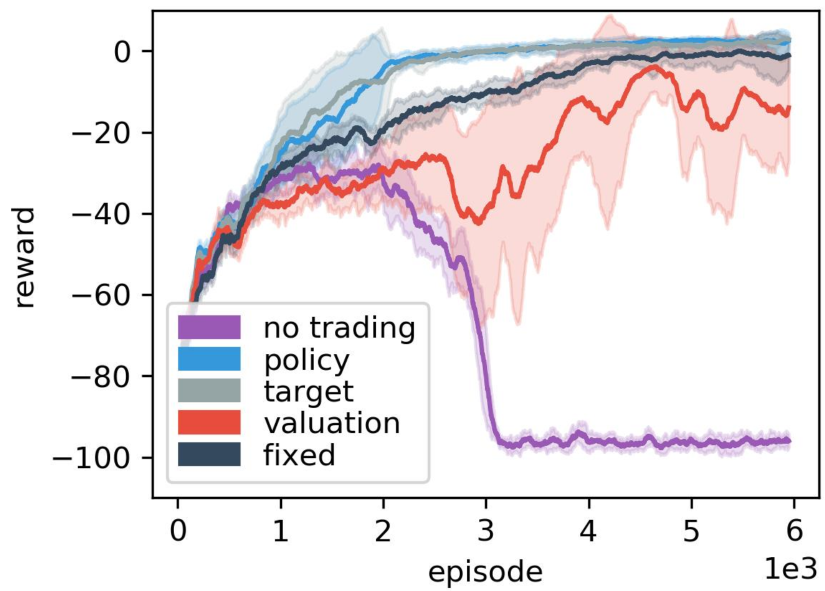


Escape Room fixed wall agent 1



Smart Factory vs Escape Room





Conclusion

Agents with action trading outperform non-trading agents across the board

Policy and Target networks performance depends strongly on learning success

Valuation networks have the worst results as they are based on non-cooperative agents

Fixed compensation achieves consistently good results, but requires manual adjustment.

Conditioning of all trading compensations

Thank you for your attention