Multi-Agent Reinforcement Learning for MAPF

Yudong Luo

`yudong.luo@uwaterloo.ca`

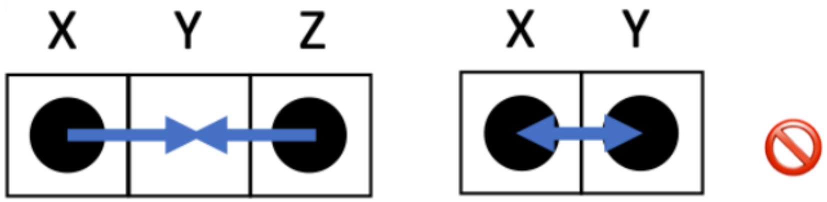
Content

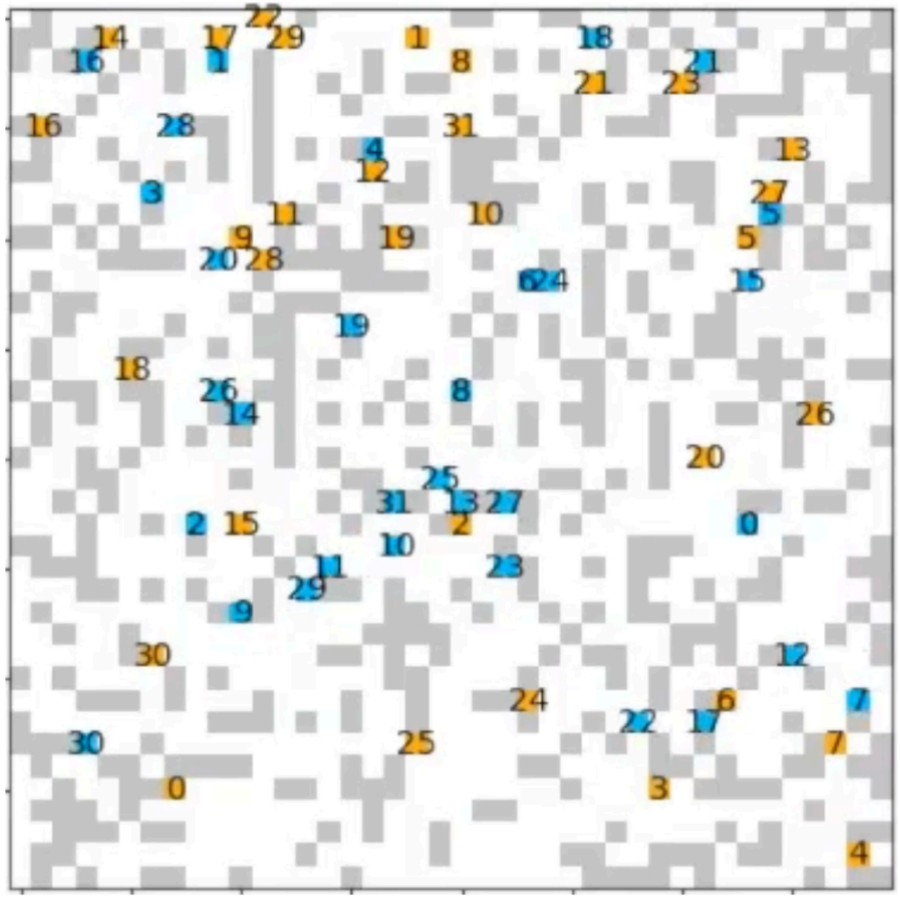
- Background: MAPF & Multi-Agent Sequential Decision Making
- Cooperative Multi-Agent Reinforcement Learning
- MAPF with deep Reinforcement Learning
- Future Perspectives

MAPF Problem

- Graph $G = (Vertices, Edges)$
- A set of N agents
- Path p_i from start to goal location
- $\min \sum_{i=1}^{N} delay(p_i)$

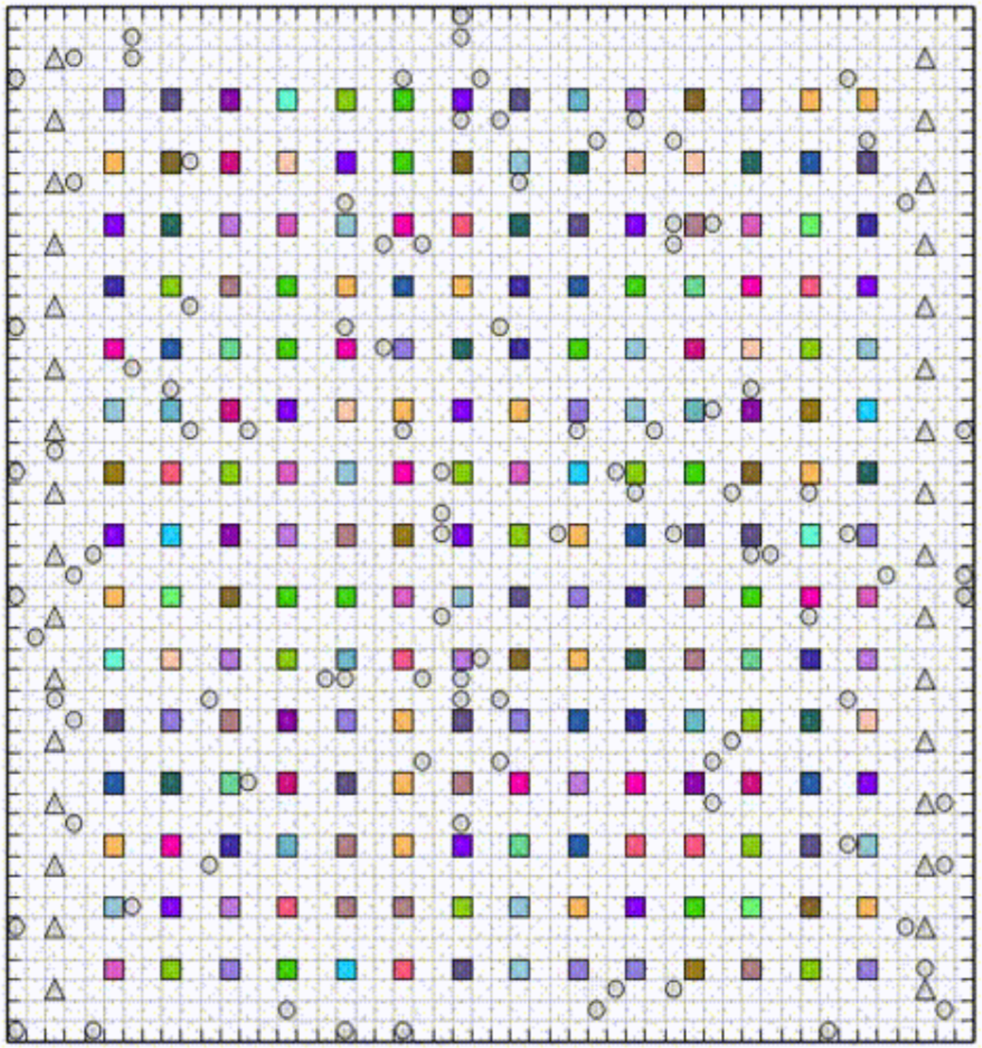
Assume: No vertex and edge collision



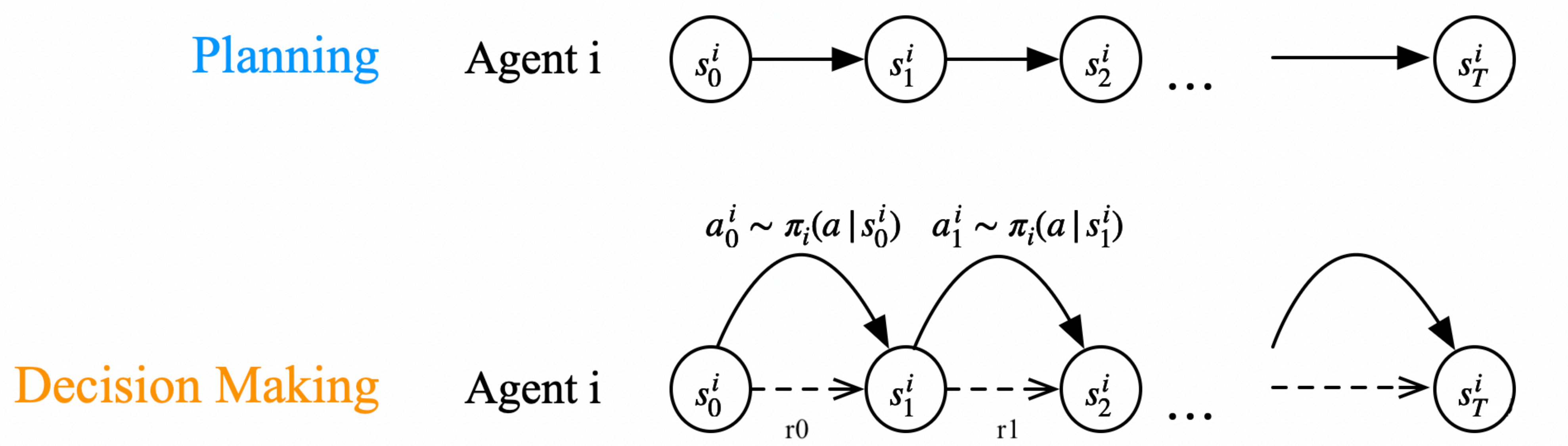


https://www.youtube.com/watch?v=1i0zNqoGRWY

Warehouse Robots



From Planning to Sequential Decision Making



$r^i(s_t^i, a_t^i)$ tells how good is the action a_t^i at s_t^i

$$\max \sum_{t=0}^{T-1} r^i(s_t^i, a_t^i)$$

Planning v.s. Reinforcement Learning (RL)

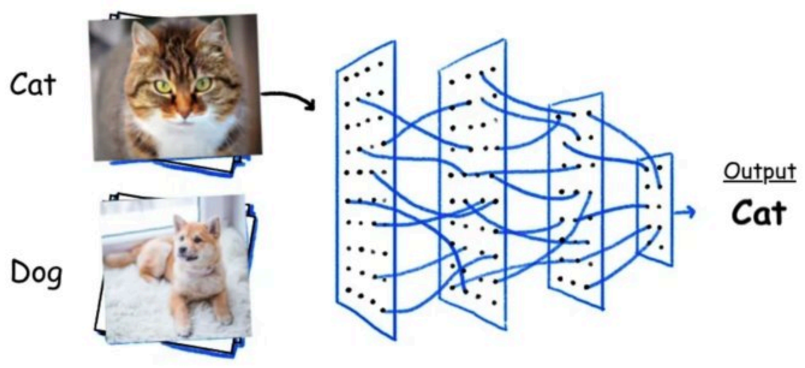
- Planning, e.g., Conflict based search (Sharon et al., 2015), uses global information
  - ✅ Pros: Optimality
  - ❌ Cons: Scalability; Efficiency
- RL, e.g., learns a function $\pi(a|s)$ to tell what action a to take at a state s, makes local decision
  - ✅ Pros: Scalability; Efficiency during execution
  - ❌ Cons: May hard to learn

Content

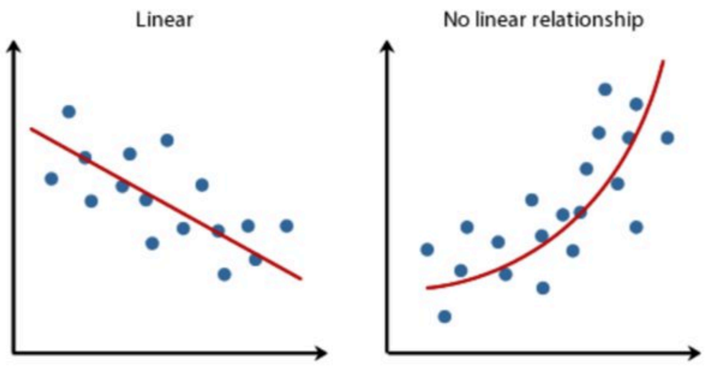
- 👉 Background: MAPF & (Multi-Agent) Sequential Decision Making
- 👉 Cooperative Multi-Agent Reinforcement Learning
  - 🖋 Single agent RL Recap
  - 🖋 From Single-Agent to Multi-Agent RL
- 👉 MAPF with deep Reinforcement Learning
- 👉 Future Perspectives

Characteristics of RL

Machine Learning, e.g., supervised Learning



Classification



Regression

- $f_\theta(\vec{x}) \quad (\theta = \{\vec{\alpha}, b\})$, e.g, $\vec{\alpha}^\top \vec{x} + b$
  - Make it Non-linear: stack linear and non-linear layers
- Update parameters: Fit many $(\vec{x}, y)$ to update θ. $\quad \min \ (\vec{\alpha}^\top \vec{x} + b - y)^2$

Characteristics of RL (Continue)

What makes RL different?

- No label (no supervisor), only a reward signal (given by the environment)
- Time really matters (sequential, non i.i.d data)
- Agent's actions affect the subsequent data it receives
- Feedback is delayed

Single agent RL

$$\max \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t R_t\Big], \quad \gamma \in (0,1)$$

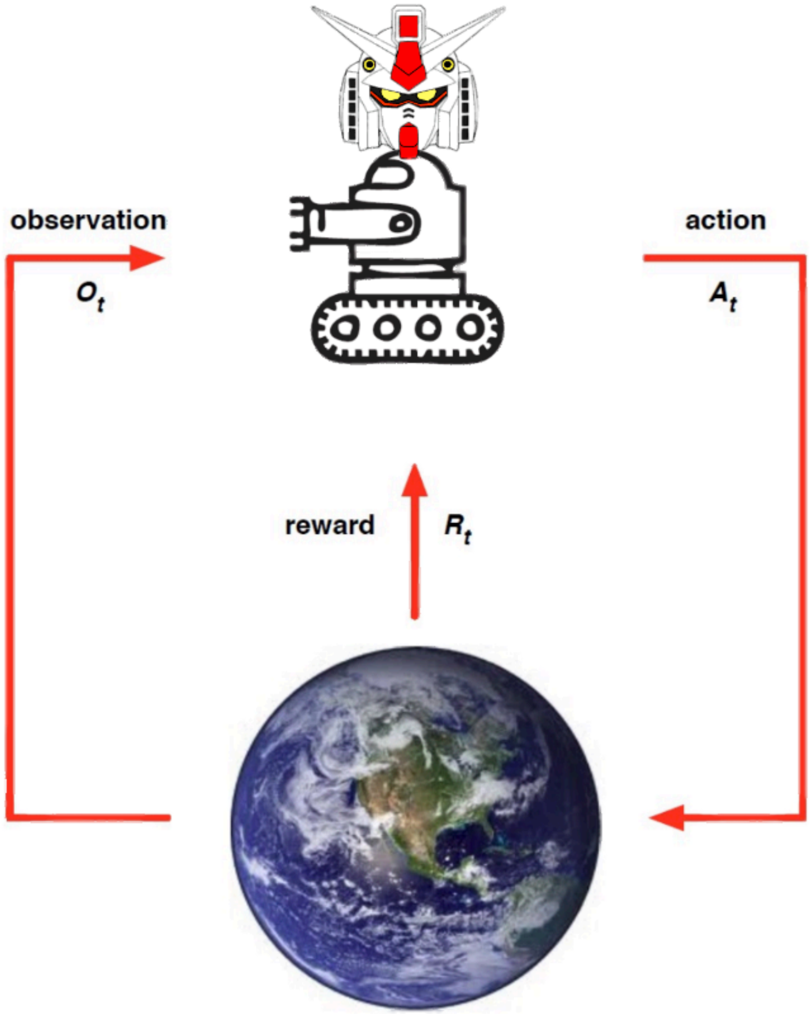
Major Components

- Policy π: mapping s to a
    - $a = \pi(s)$ or $a \sim \pi(\cdot|s)$
- Value function (goodness/badness of states)

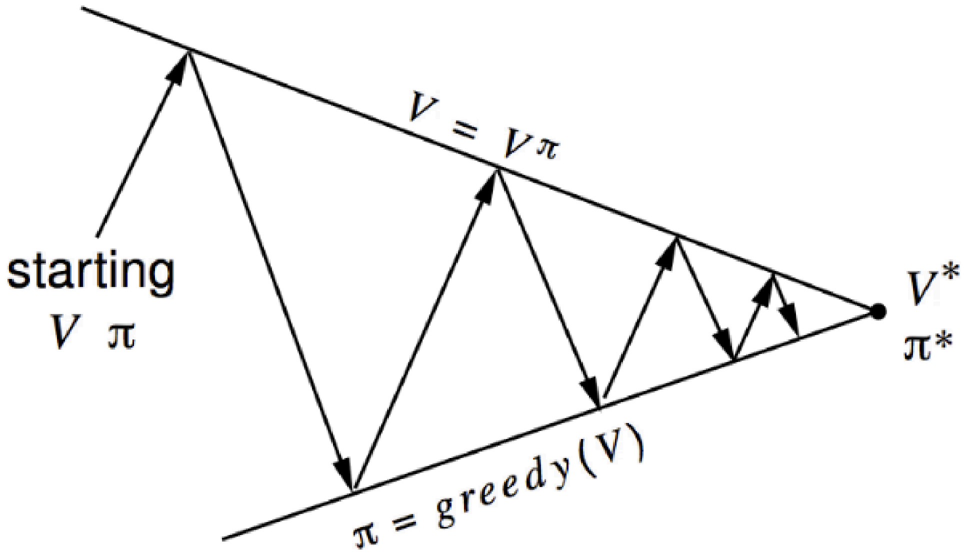
$$V^\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \dots | S_t = s]$$

$$Q^\pi(s,a) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \dots | S_t = s, A_t = a]$$

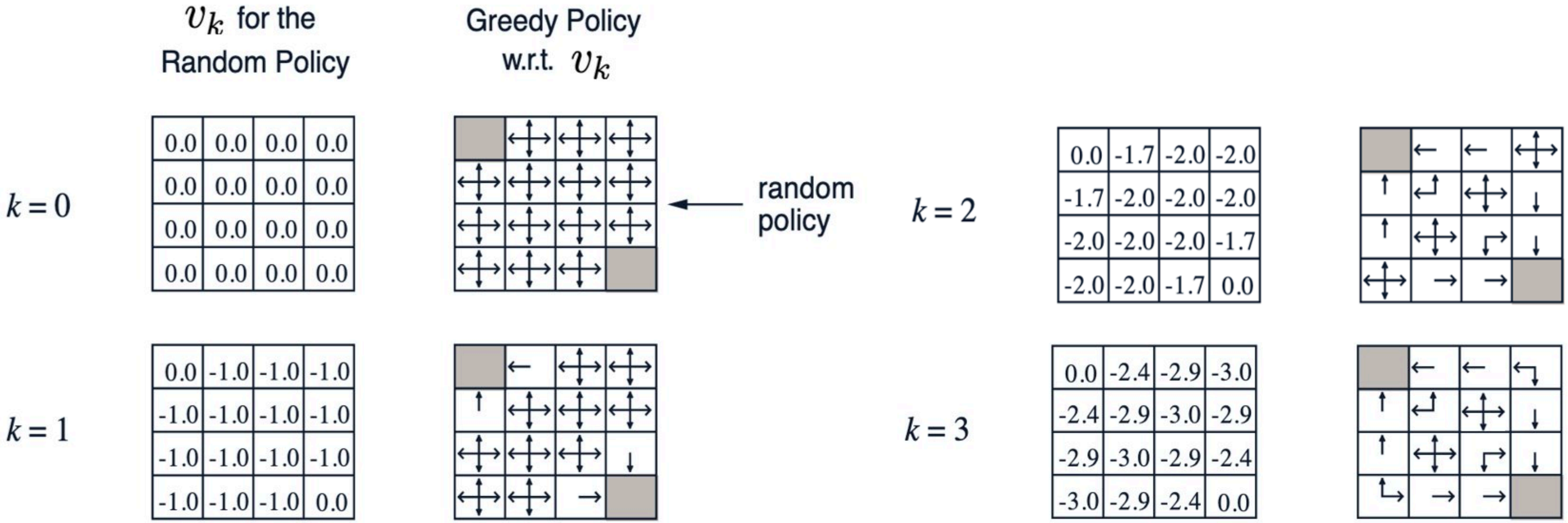
$$V^\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma V^\pi(S_{t+1}) | S_t = s]$$



Find Optimal Policy: Value/Policy Iteration

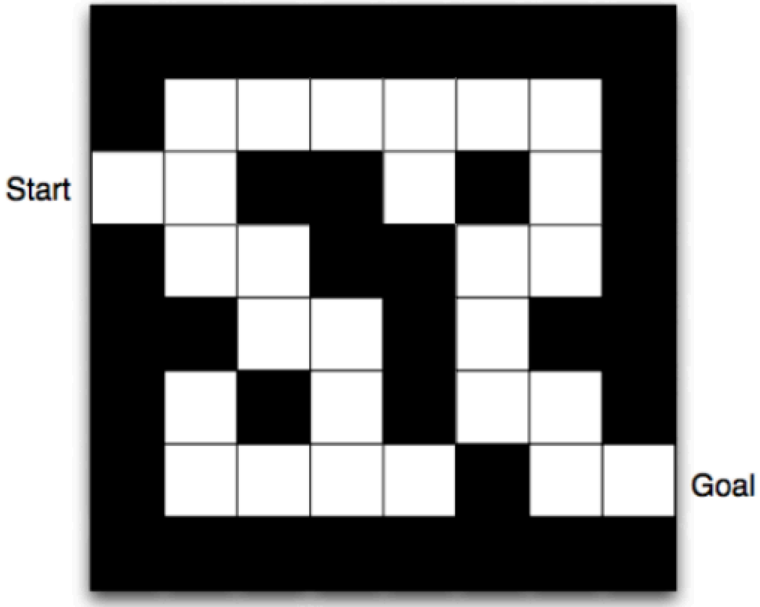


Value/Policy Iteration



$$V^{\pi}(s) = \mathbb{E}_{\pi}[R_{t+1} + \gamma V^{\pi}(S_{t+1}) | S_t = s]$$

Reward function



Objective in RL:

$\max \mathbb{E}\Big[\sum_t \gamma^t R(s_t, a_t)\Big]$

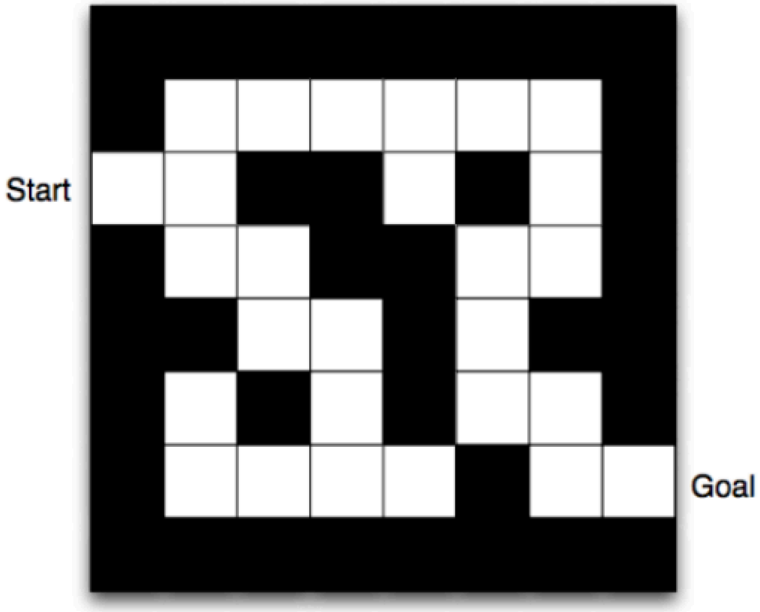
Objective in Path Finding:

min total steps (max $-1\times$(total steps))

Reward:

$r(s_t, a_t) = -1$

Reward function



Objective in RL:

$\max \mathbb{E}\left[\sum_t \gamma^t R(s_t, a_t)\right]$

Objective in Path Finding:

min total steps (max $-1\times$(total steps))

Reward:

$r(s_t, a_t) = -1$

Can we set the reward as ($\gamma = 0.99$)

- 0 for each step, 1 for reaching the goal?

Prominent RL Algorithms

Value-based

- $\pi^*(s) = \arg\max_a Q^*(s, a)$
- Q-learning (Sutton & Barto, 1998), Double Q-learning (Hasselt, 2010)

Policy gradient-based

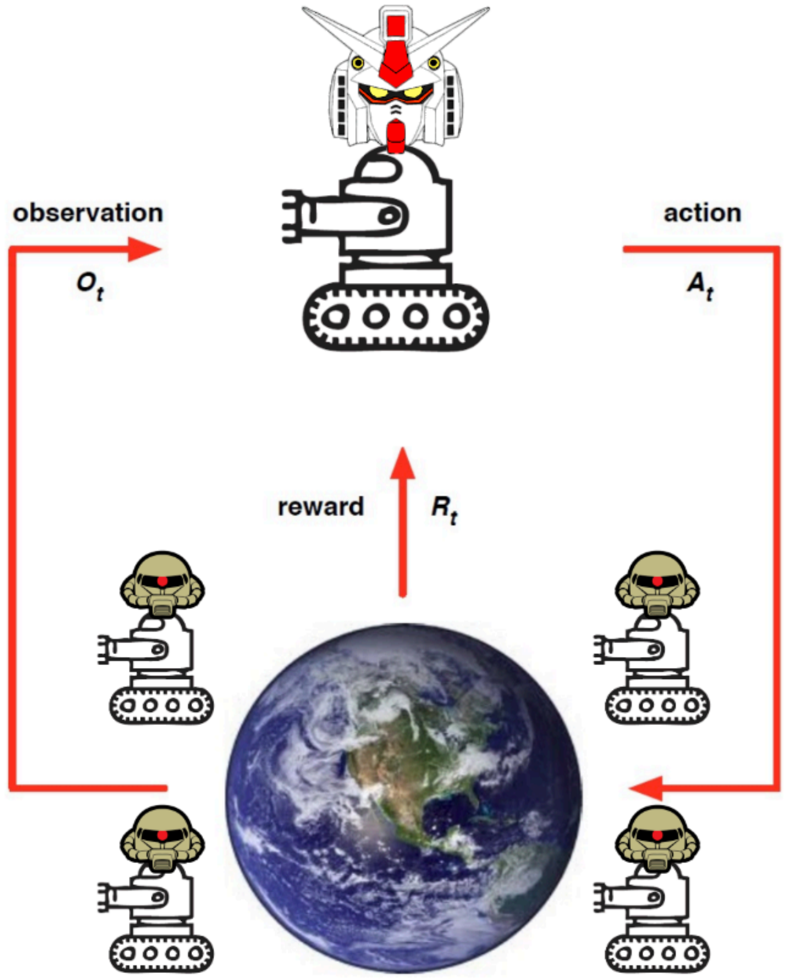
- update π towards higher value of $Q(s, a)$
- Asynchronous advantage actor-critic (A3C) (Mnih et al., 2016)
- Deep deterministic policy gradient (DDPG) (Lillicrap et al., 2016)
- Proximal policy optimization (PPO) (Schulman et al., 2017)
- Soft actor-critic (SAC) (Haarnoja et al., 2018)

Content

- Background: MAPF & (Multi-Agent) Sequential Decision Making
- Cooperative Multi-Agent Reinforcement Learning
  - Single agent RL Recap
  - From Single-Agent to Multi-Agent RL
- MAPF with deep Reinforcement Learning
- Future Perspectives

Multi-Agent RL

- Learn from interaction with the environment (exploration)
- The environment contains other agents that are learning and updating (Non-stationary)

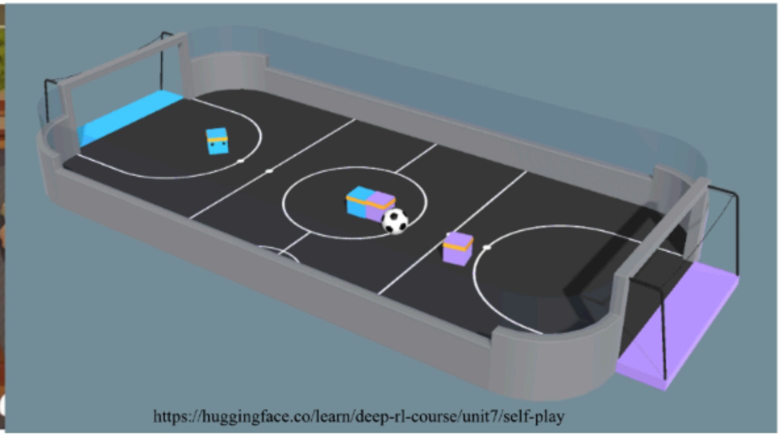


Scenarios

Cooperative

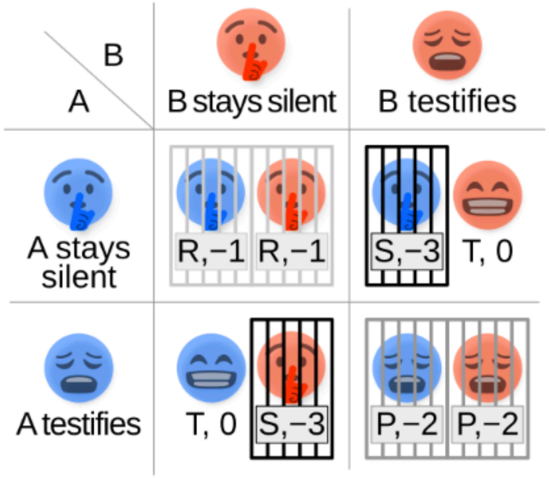
Overcook

Competitive



sports

Mixed motive



prisoner's dilemma

Cooperative game

$$\max \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \mathbf{a}_t)\right], \quad R(s_t, \mathbf{a}_t) = \sum_{i=1}^{N} R_i(s_t, \mathbf{a}_t)$$

How to learn optimal policies when we have multiple agents?

Cooperative game

$$\max \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t R(s_t, \mathbf{a}_t)\Big], \quad R(s_t, \mathbf{a}_t) = \sum_{i=1}^{N} R_i(s_t, \mathbf{a}_t)$$

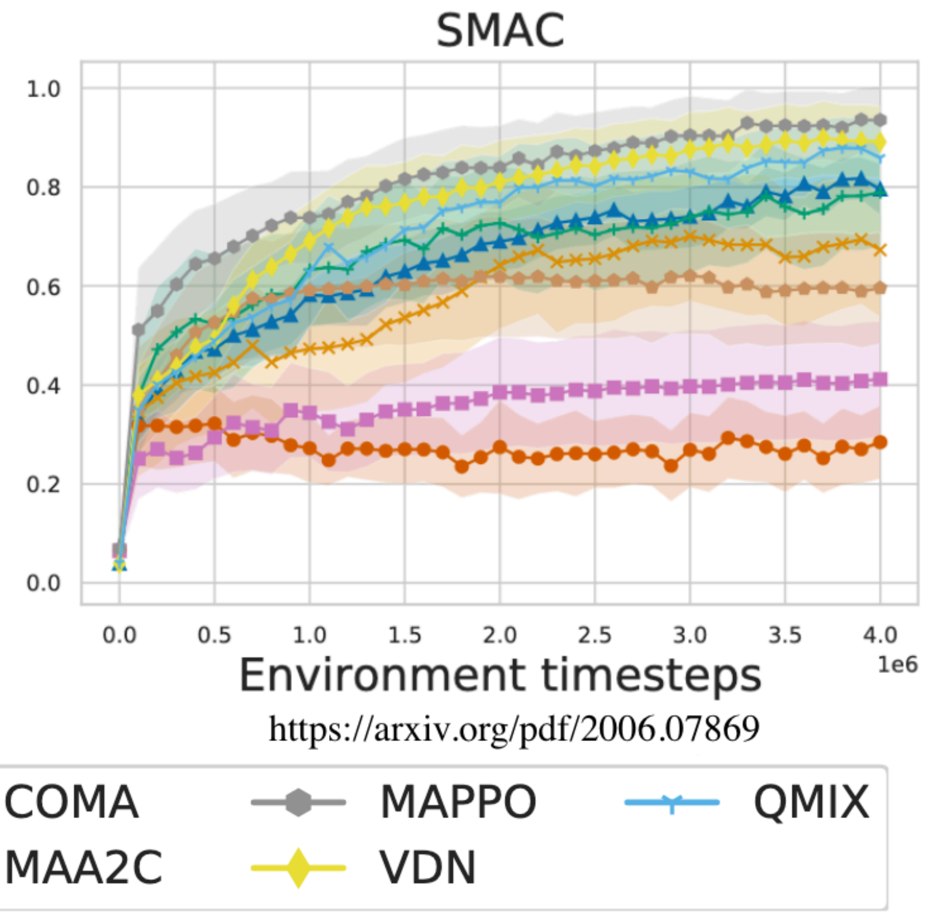
$$\max \Bigg(\underbrace{\mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t R_1(s_t, \mathbf{a}_t)\Big]}_{\text{Agent } 1} + \underbrace{\mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t R_2(s_t, \mathbf{a}_t)\Big]}_{\text{Agent } 2} + \ldots + \underbrace{\mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t R_N(s_t, \mathbf{a}_t)\Big]}_{\text{Agent } N}\Bigg)$$

Can we do independent learning?

In practice, yes

- The StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al., 2019)

https://arxiv.org/pdf/1902.04043

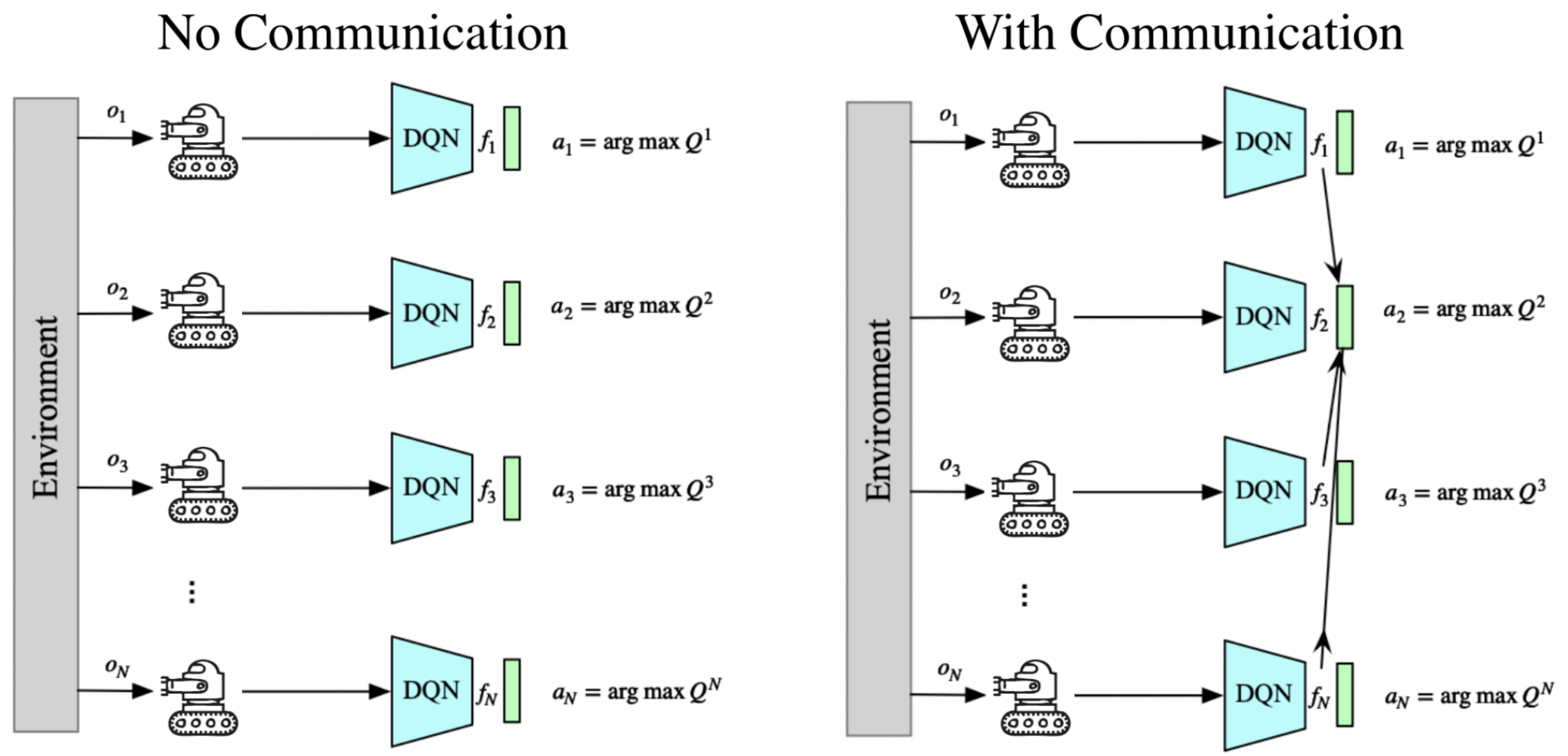


https://arxiv.org/pdf/2006.07869

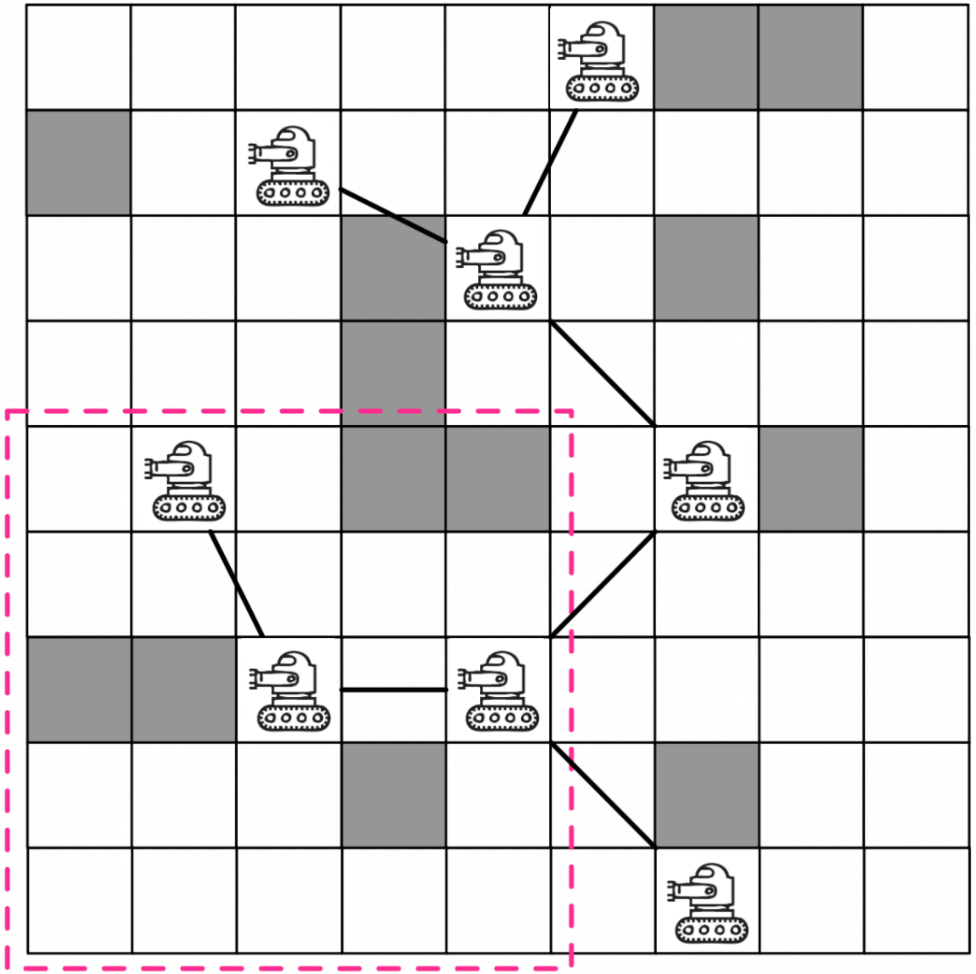
Coordination of Agents

- Communication between agents
  - Build local communication channels between agents via hidden features
  - TarMAC (Das et al., 2019), DGN (Jiang et al., 2020) etc.
- Centralized training & decentralized execution
  - Train a centralized value function to guide the update of each policy. Execute the policies in a decentralized way.
  - MADDPG (Lowe et al., 2017), QMIX (Rashid et al., 2018) etc.
- Opponent modeling
  - Observe and predict the actions of other agents, so as to perform accordingly
  - ROMMEO (Tian et al., 2019), PR2 (Wen et al., 2019) etc.

Communication

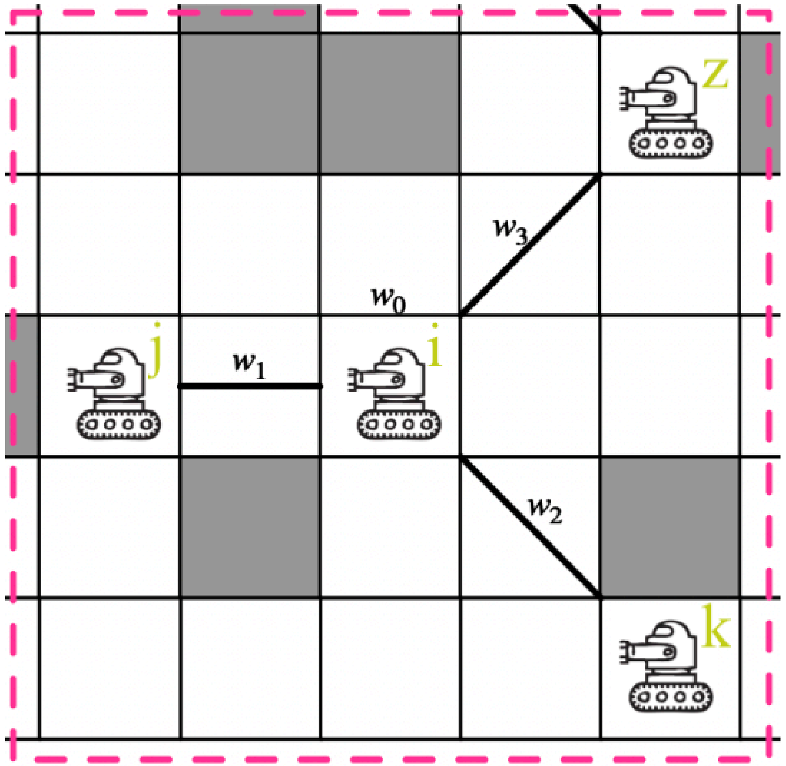


Graph Convolution with Attention

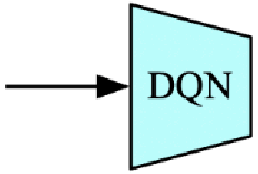


Attention

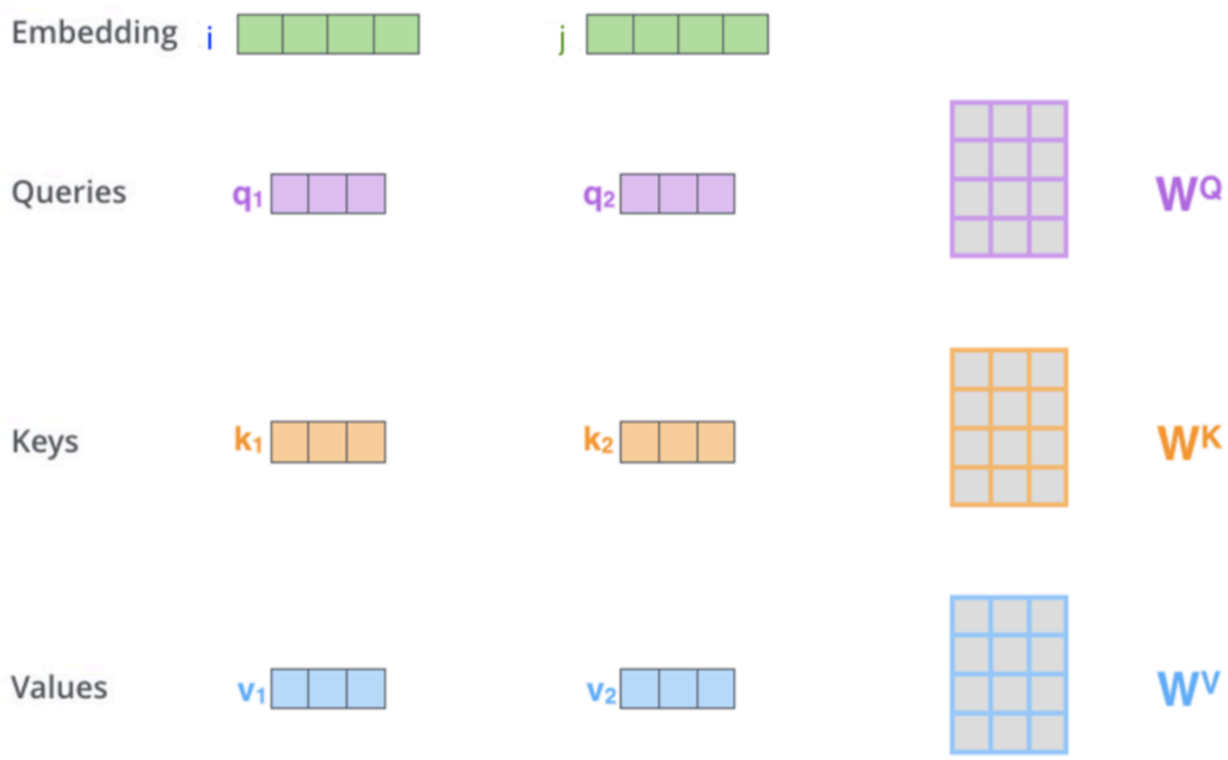
At a high level

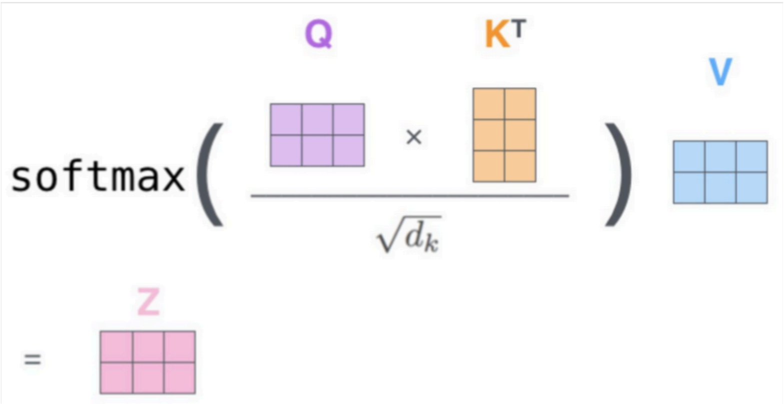


$$
\begin{aligned}
\hat{f}_i = & w_0 \times f_i + \\
& w_1 \times f_j + \\
& w_2 \times f_k + \\
& w_3 \times f_z \\
& w_0 + \ldots + w_3 = 1
\end{aligned}
$$



Attention (Continue)





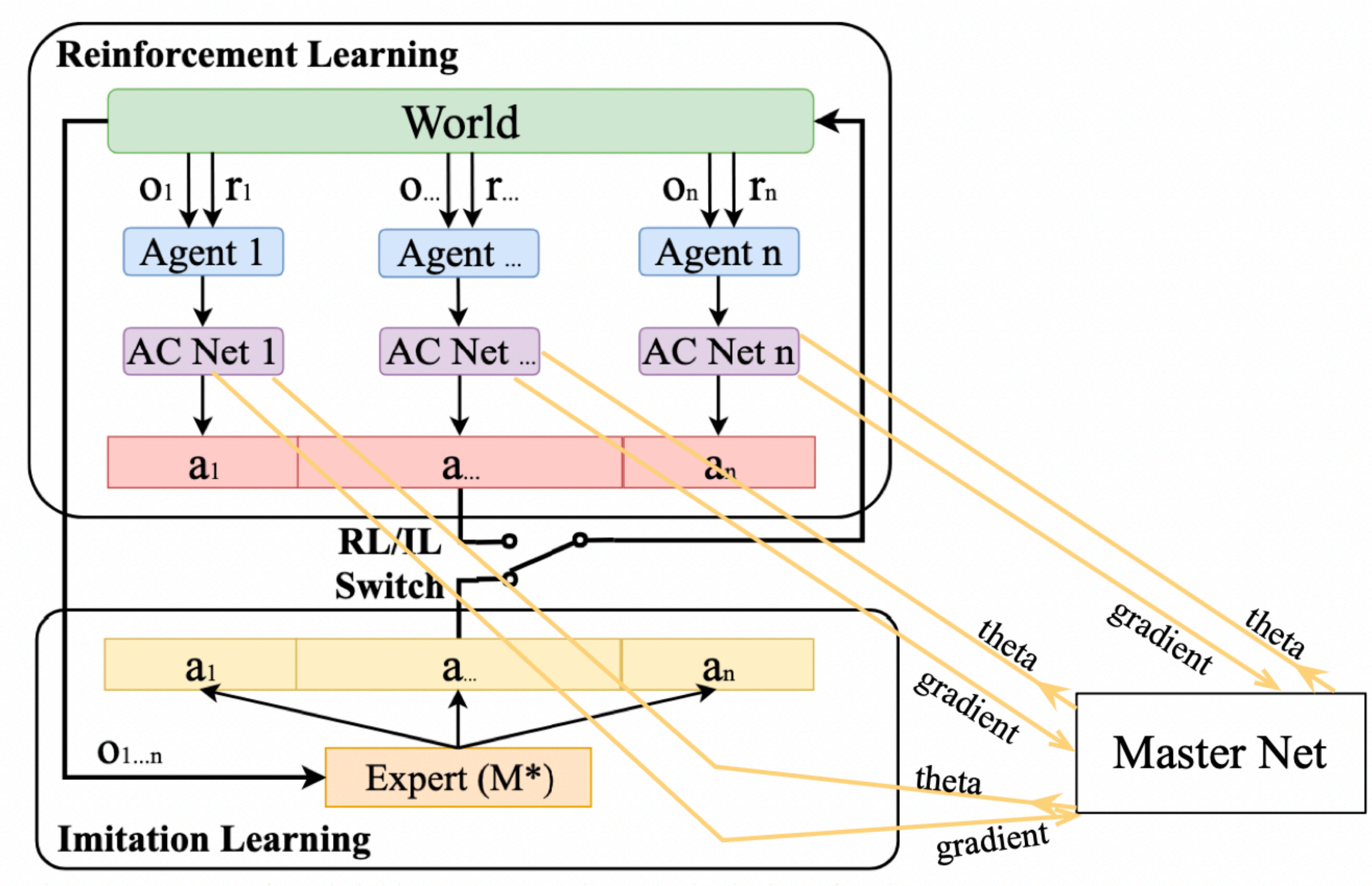
W^Q, W^K, W^V are learning parameters.

Content

- 👉 Background: MAPF & (Multi-Agent) Sequential Decision Making
- 👉 Cooperative Multi-Agent Reinforcement Learning
- 👉 MAPF with deep Reinforcement Learning
- 👉 Future Perspectives

PRIMAL

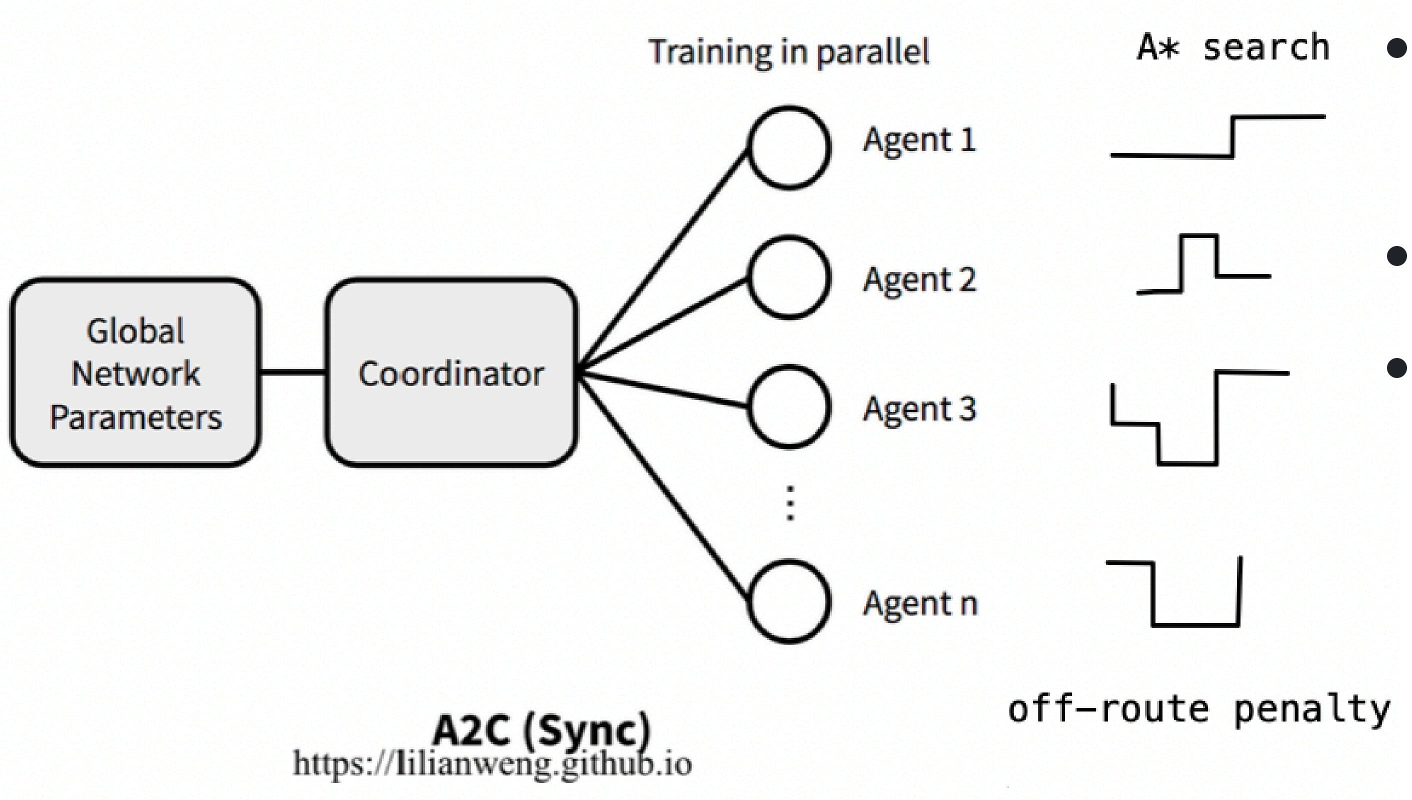
MAPF via Mulai-Agent Reinforcement and Imitation Learning (Sartoretti et al., 2019)



- A3C as the backend
- Each π_i synchronous with master π
- Switch between RL and Imitation (supervise)

MAPPER

MAPF via RL with off-route penalty (Liu et al., 2020)



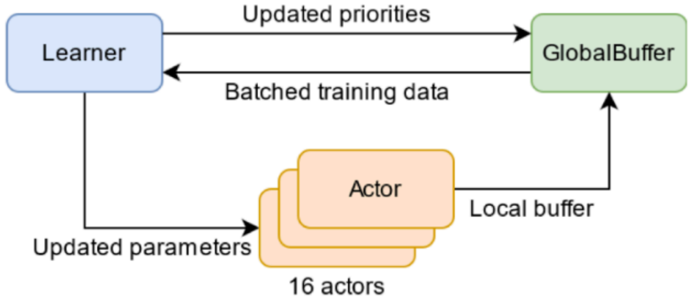
- A2C (Mnih et al., 2016) as the backend
- A* Planning
- Off-route penalty reward

Globally Guided Reinforcement Learning for MAPF (Wang et al., 2020)

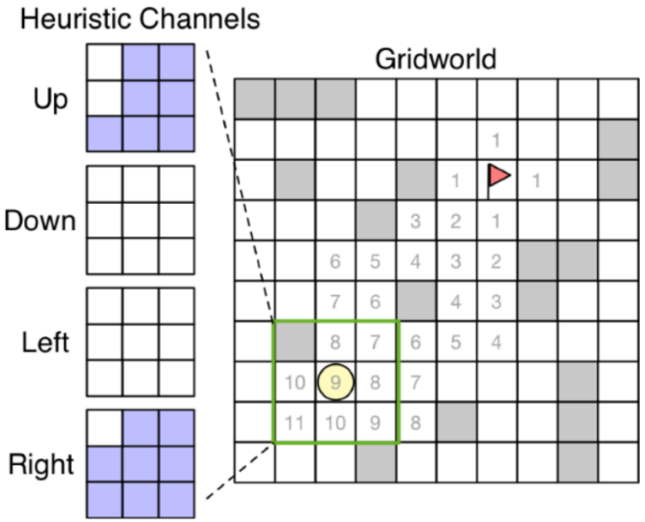
DHC

Ma et al. (2021)

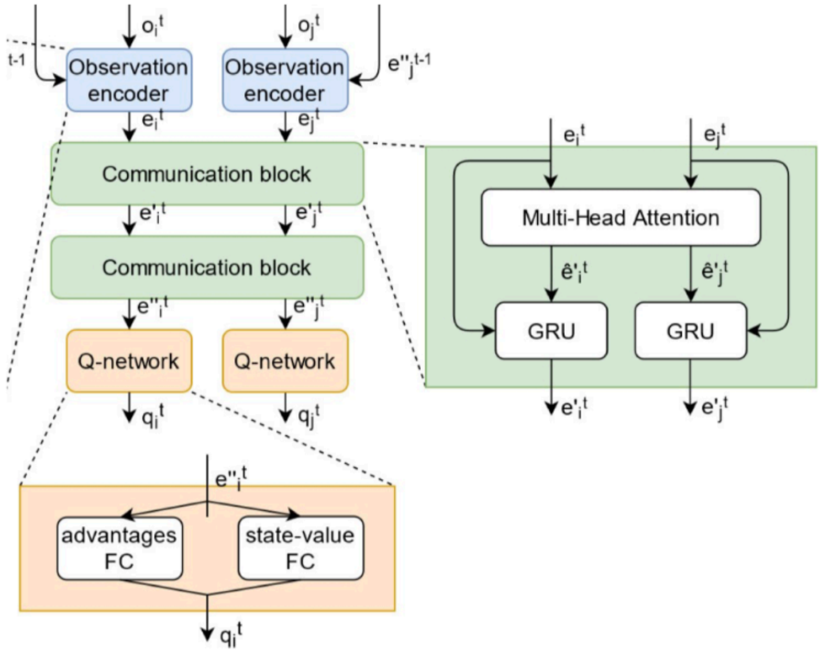
Distributed



Heuristic



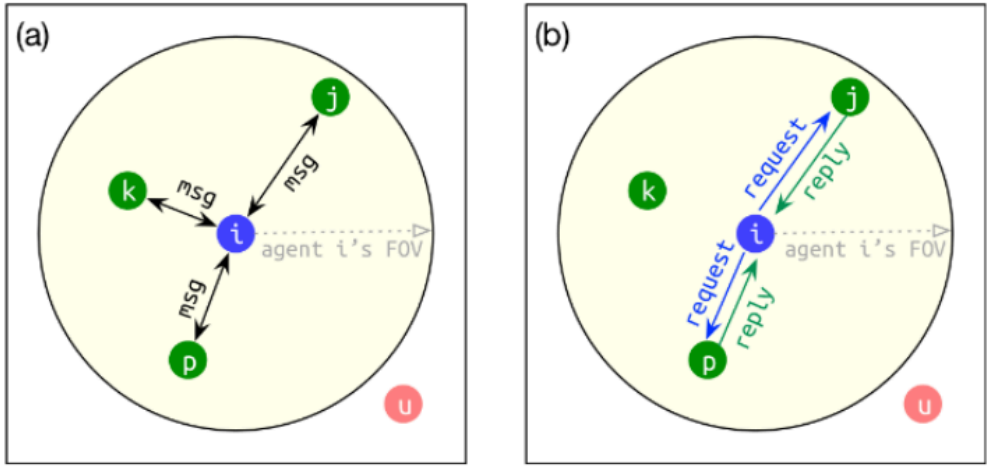
Communication



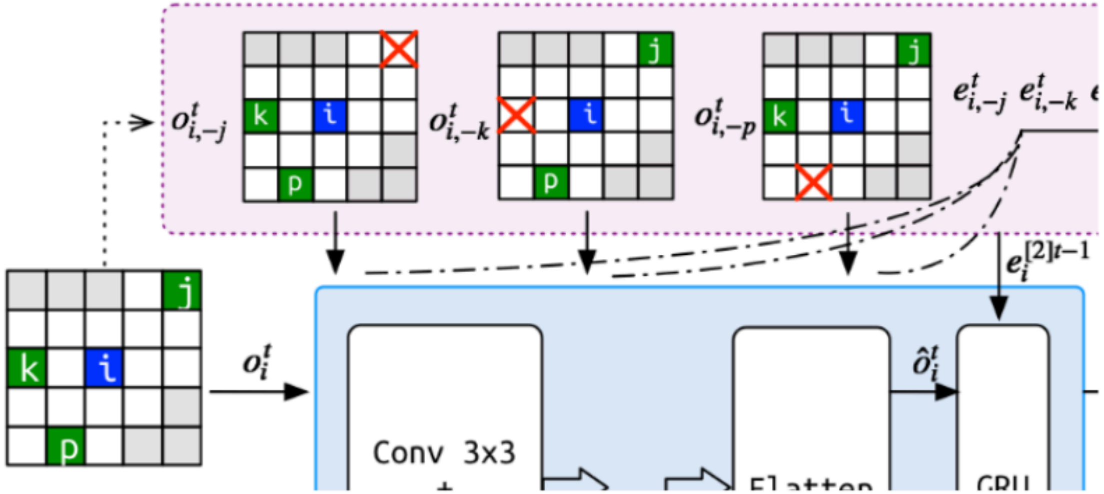
DCC

Decision Causal Communication (Ma et al., 2021)

Broadcast v.s. Request-reply

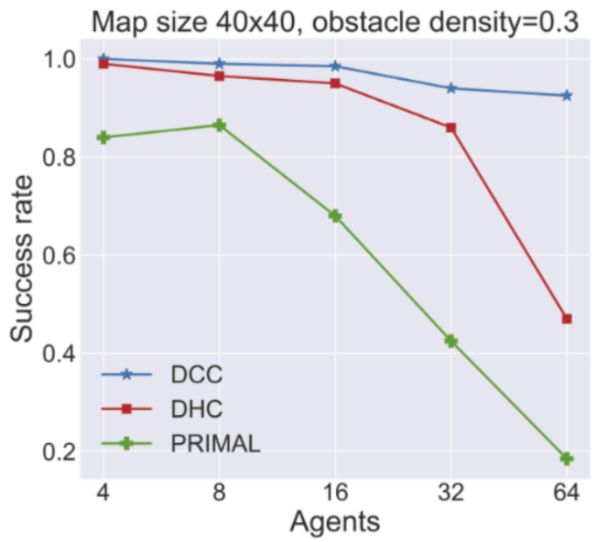


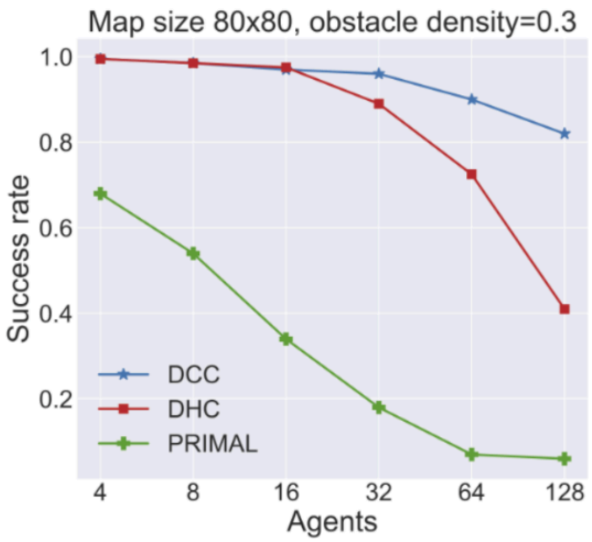
DCC



DCC (Continue)

Performance





Communication frequency

	Map size 40×40		Map size 80×80	
Agents	DCC	RR-N2	DCC	RR-N2
4	2.42	36.88	1.06	18.36
8	11.56	209.79	5.75	105.86
16	60.47	959.38	24.98	469.58
32	294.69	4111.57	126.685	2125.94
64	1811.33	19490.09	562.11	8780.72
128	-	-	2915.84	36560.30

RR-N2: Request-reply with nearest 2 agents

Demo

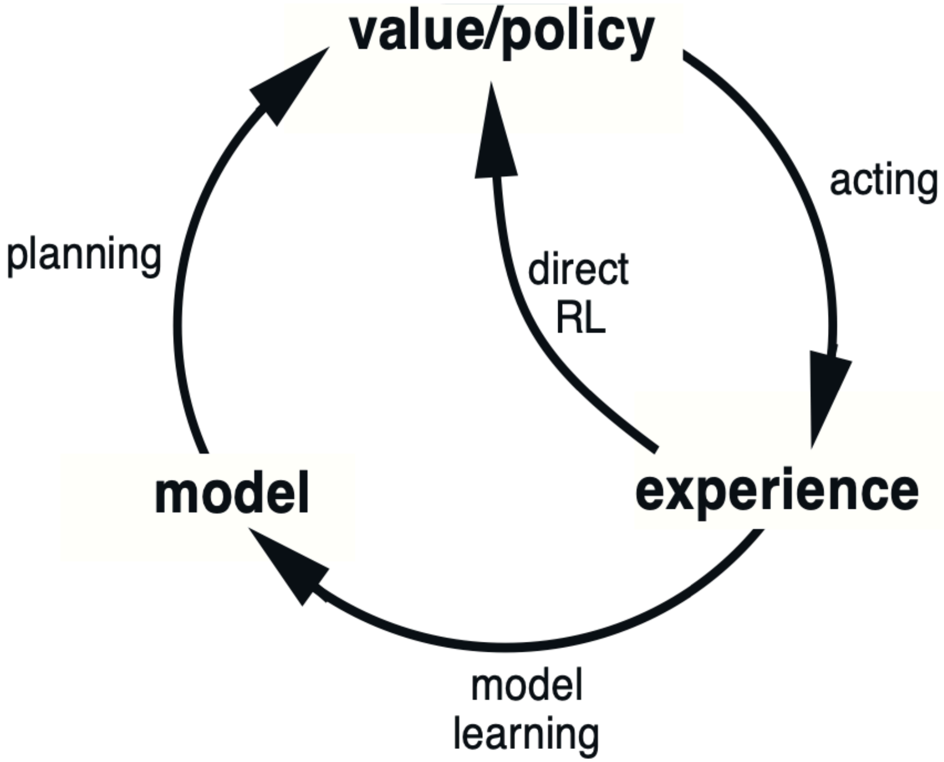
https://www.youtube.com/watch?v=1i0zNqoGRWY

https://www.youtube.com/watch?v=ZinvpFgMlGs

Content

- Background: MAPF & (Multi-Agent) Sequential Decision Making
- Cooperative Multi-Agent Reinforcement Learning
- MAPF with deep Reinforcement Learning
- Future Perspectives

A. Model-based RL



http://incompleteideas.net/book/RLbook2020.pdf

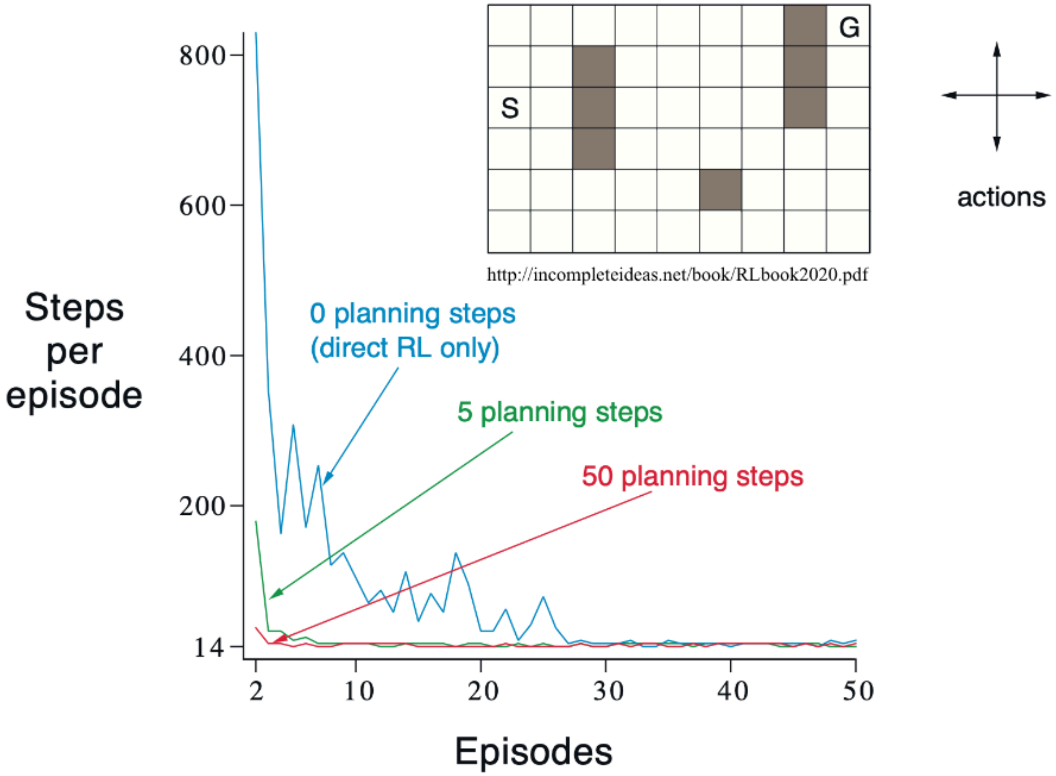
- Model-free (unaware of the env)
- Exploration is expensive
- Learning is slow

- The transition dynamic of MAPF problem is usually not complex

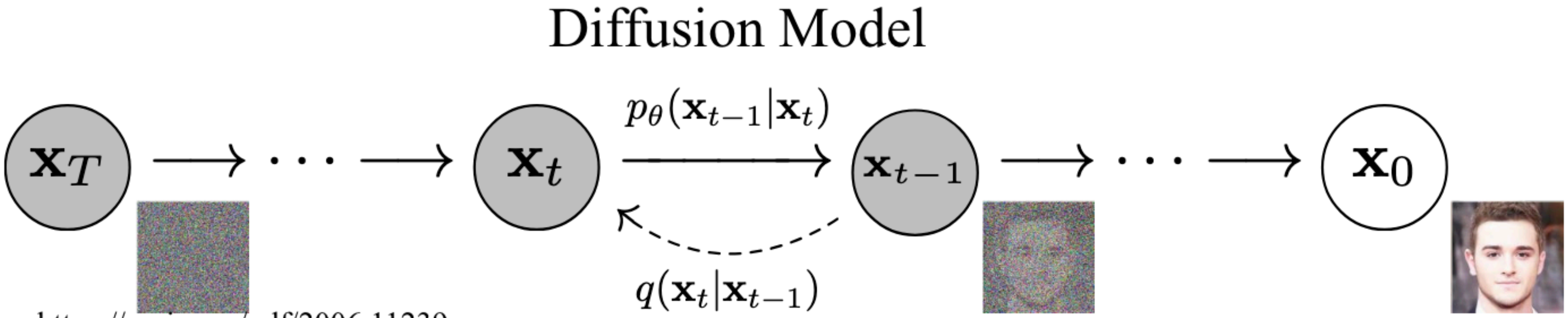
Dyna-Q

Tabular Dyna-Q

Initialize $Q(s,a)$ and $Model(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$
Loop forever:
(a) $S \leftarrow$ current (nonterminal) state
(b) $A \leftarrow \varepsilon$-greedy(S,Q)
(c) Take action A; observe resultant reward, R, and state, S'
(d) $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
(e) $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment)
(f) Loop repeat n times:
$S \leftarrow$ random previously observed state
$A \leftarrow$ random action previously taken in S
$R, S' \leftarrow Model(S,A)$
$Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$



B. Generative Modeling

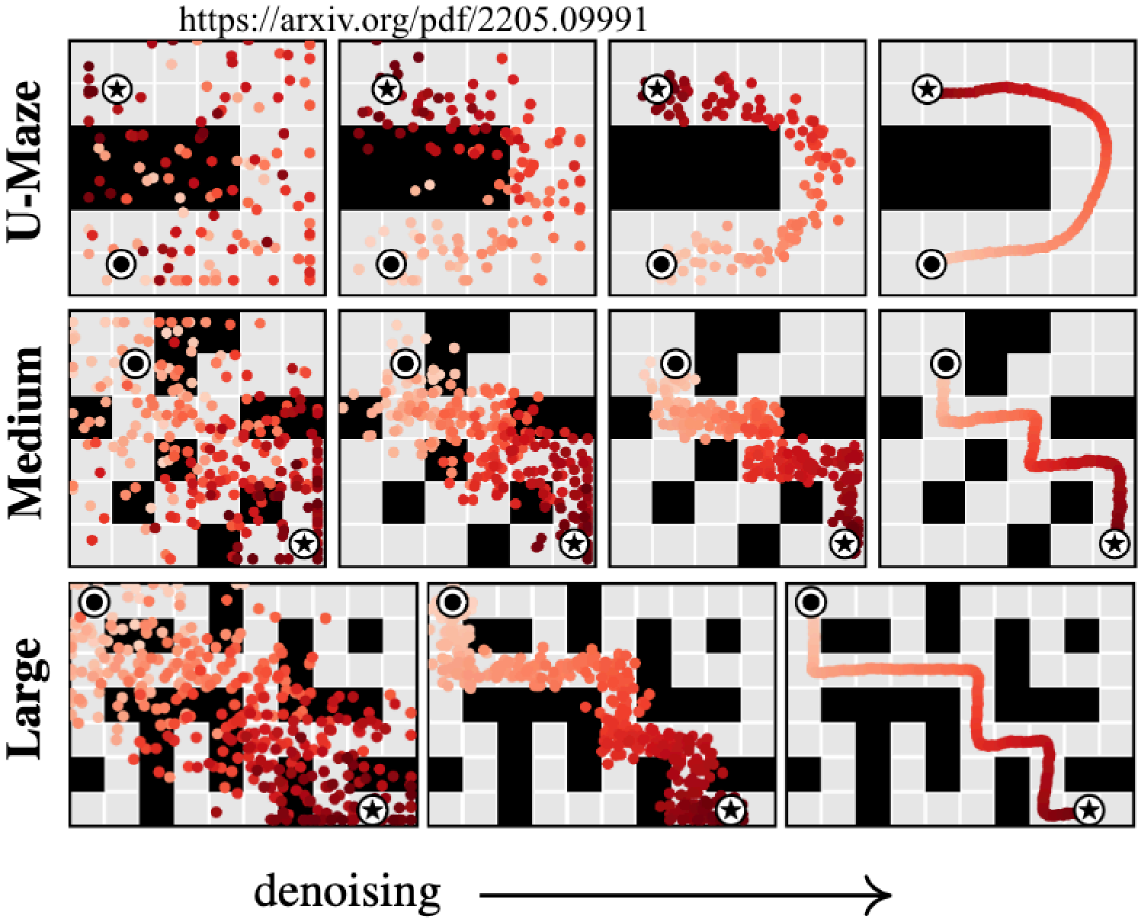


https://arxiv.org/pdf/2006.11239

- Learn a distribution $p_\theta(x|z), z \sim \mathcal{N}(0,1)$ to fit data distribution $\mathcal{D}_x$
- Very powerful and expressive

Diffusion for Planning

Diffuser (Janner et al., 2022), Conditional Decision Diffuser (Ajay et al., 2023)



References

Sharon, Guni, et al. "Conflict-based search for optimal multi-agent pathfinding." Artificial intelligence 219 (2015): 40-66.

Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. Vol. 1. No. 1. Cambridge: MIT press, 1998.

Hasselt, Hado. "Double Q-learning." Advances in neural information processing systems 23 (2010).

Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." nature 518.7540 (2015): 529-533.

Van Hasselt, Hado, Arthur Guez, and David Silver. "Deep reinforcement learning with double q-learning." Proceedings of the AAAI conference on artificial intelligence. Vol. 30. No. 1. 2016.

Mnih, Volodymyr, et al. "Asynchronous methods for deep reinforcement learning." International conference on machine learning. PmLR, 2016.

Lillicrap, Timothy P., et al. "Continuous control with deep reinforcement learning." ICLR (2016).

Schulman, John, et al. "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347 (2017).

Haarnoja, Tuomas, et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor." International conference on machine learning. Pmlr, 2018.

Samvelyan, Mikayel, et al. "The starcraft multi-agent challenge." arXiv preprint arXiv:1902.04043 (2019).

Das, Abhishek, et al. "Tarmac: Targeted multi-agent communication." International Conference on machine learning. PMLR, 2019.

Jiang, Jiechuan, et al. "Graph convolutional reinforcement learning." ICLR (2020).

Lowe, Ryan, et al. "Multi-agent actor-critic for mixed cooperative-competitive environments." Advances in neural information processing systems 30 (2017).

Rashid, Tabish, et al. "QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning." ICML (2018).

Tian, Zheng, et al. "A regularized opponent model with maximum entropy objective." IJCAI (2019).

Wen, Ying, et al. "Probabilistic recursive reasoning for multi-agent reinforcement learning." ICLR (2019).

Sartoretti, Guillaume, et al. "Primal: Pathfinding via reinforcement and imitation multi-agent learning." IEEE Robotics and Automation Letters 4.3 (2019): 2378-2385.

Liu, Zuxin, et al. "Mapper: Multi-agent path planning with evolutionary reinforcement learning in mixed dynamic environments." 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020.

Mnih, Volodymyr, et al. "Asynchronous methods for deep reinforcement learning." International conference on machine learning. PmLR, 2016.

Wang, Binyu, et al. "Mobile robot path planning in dynamic environments through globally guided reinforcement learning." IEEE Robotics and Automation Letters 5.4 (2020): 6932-6939.

Ma, Ziyuan, Yudong Luo, and Hang Ma. "Distributed heuristic multi-agent path finding with communication." 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021.

Ma, Ziyuan, Yudong Luo, and Jia Pan. "Learning selective communication for multi-agent path finding." IEEE Robotics and Automation Letters 7.2 (2021): 1455-1462.

Janner, Michael, et al. "Planning with diffusion for flexible behavior synthesis." ICML (2022).

Ajay, Anurag, et al. "Is conditional generative modeling all you need for decision-making?." ICLR (2023).