Multi-agent Reinforcement Learning (3): Advanced topics

Prof. Jun Wang

Computer Science, UCL

Recap

- Lecture 1: Multiagent AI and basic game theory
- Lecture 2: Potential games, and extensive form and repeated games
- Lecture 3: Solving (“Learning”) Nash Equilibria
- Lecture 4: Bayesian Games, auction theory and mechanism design
- Lecture 5: Learning and deep neural networks
- Lecture 6: Single-agent Learning (1)
- Lecture 7: Multi-agent Learning (1)
- Lecture 8: Single-agent Learning (2)
- Lecture 9: Multi-agent Learning (2)
- Lecture 10: Multi-agent Learning: advanced topics (3)

Recap on MARL(1)

- Stochastic Games
  - Policy Iteration/Value Iteration (model based)
- Equilibrium Learners (model free)
  - Nash-Q
  - Minimax-Q
  - Friend-Foe-Q
- Best-Response Learners (model free)
  - JAL and Opponent Modelling
  - Iterated Gradient Ascent
  - Wolf-IGA

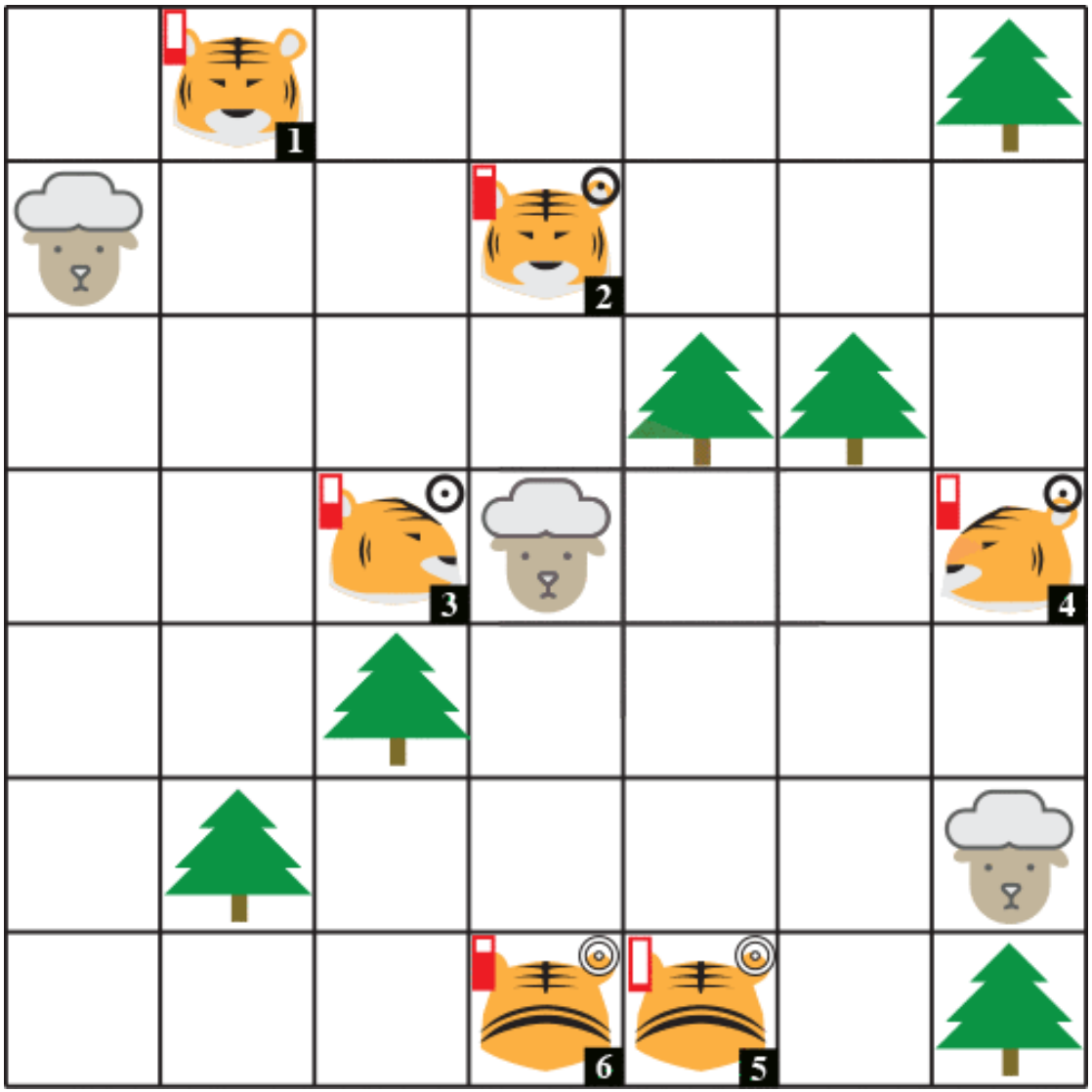
Content

- Emergent behaviours
- Agents modeling agents
- Learning communication
- Learning cooperation
- Many-agent learning

Content

- Emergent behaviours
- Agents modeling agents
- Learning communication
- Learning cooperation
- Many-agent learning

Artificial Population: Large-scale predator-prey world



The setting:

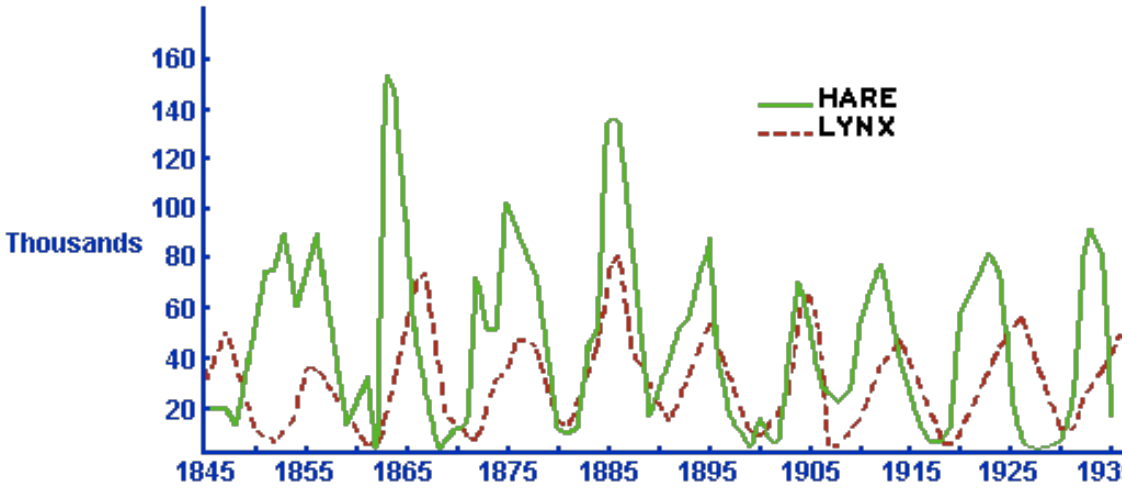
- **Predators** hunt the **prey** so as to survive from starvation.
- Each predator has its own health bar and eyesight view.
- Predators can form **a group** to hunt the prey
- Predators are scaled up to **1 million**

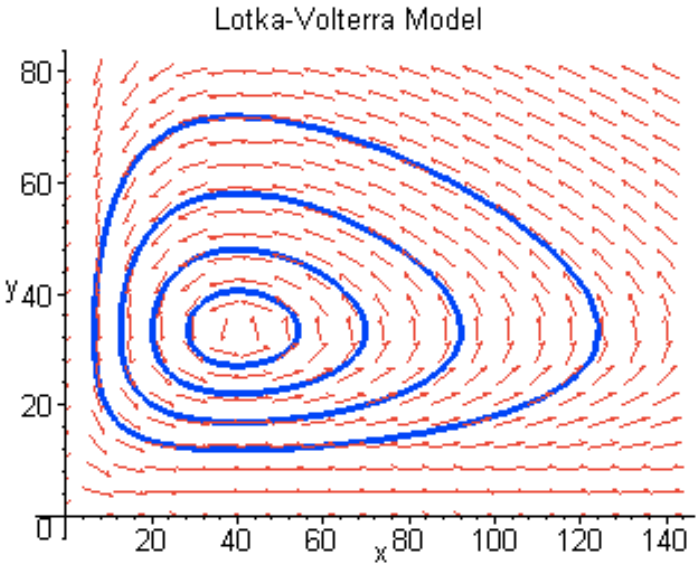
Predator Prey Obstacle Health 3 ID Group1 Group2

Yaodong Yang , Lantao Yu , Yiwei Bai , Jun Wang , Weinan Zhang , Ying Wen , Yong Yu, , Dynamics of Artificial Populations by Million-agent Reinforcement Learning, 2017

Ecology: the Lotka-Volterra (LV) model

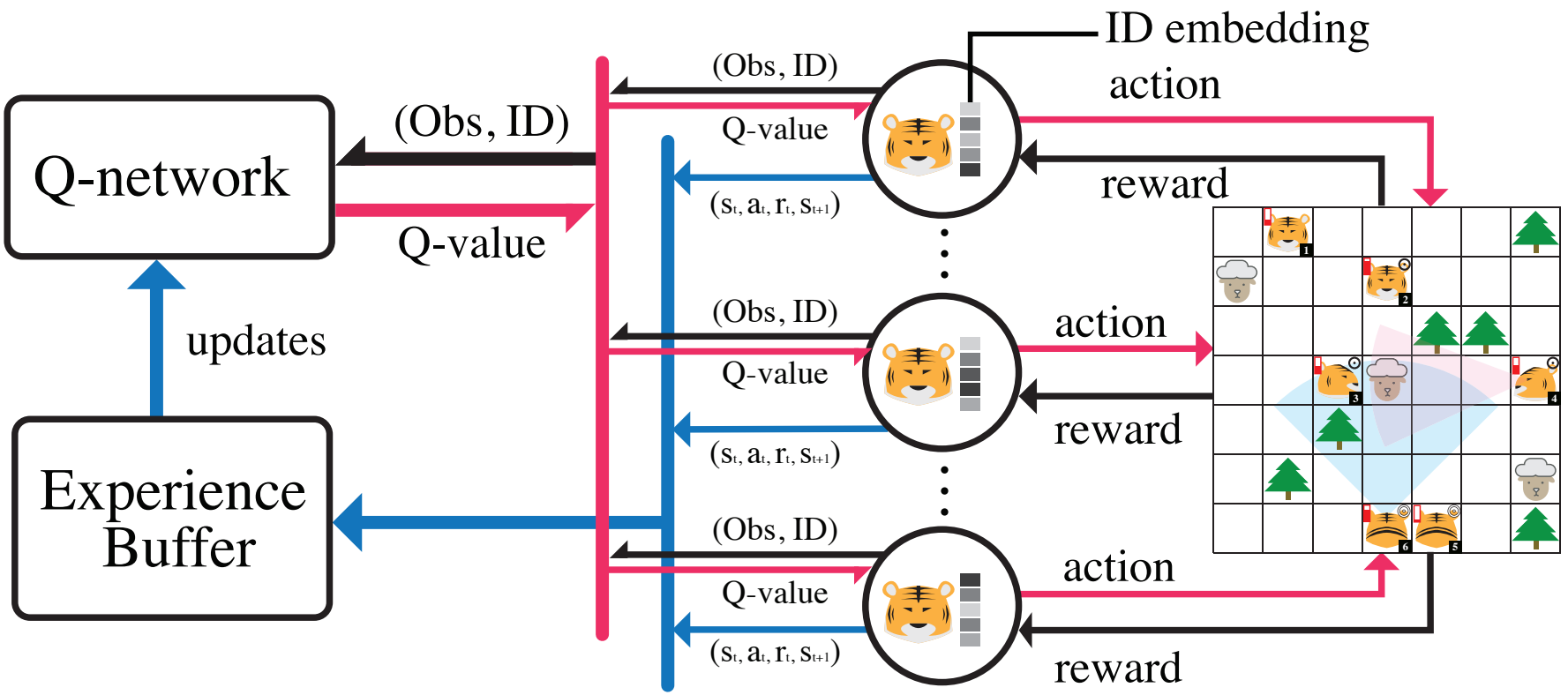
- A major topic of population dynamics is the cycling of predator and prey populations
- The *Lotka-Volterra* model is used to model this
- lynx (wild cat) and hare





Lotka, A. J. (1910). "Contribution to the Theory of Periodic Reaction". *J. Phys. Chem.* **14** (3): 271–274.

Reinforcement Learning with 1 millions agents

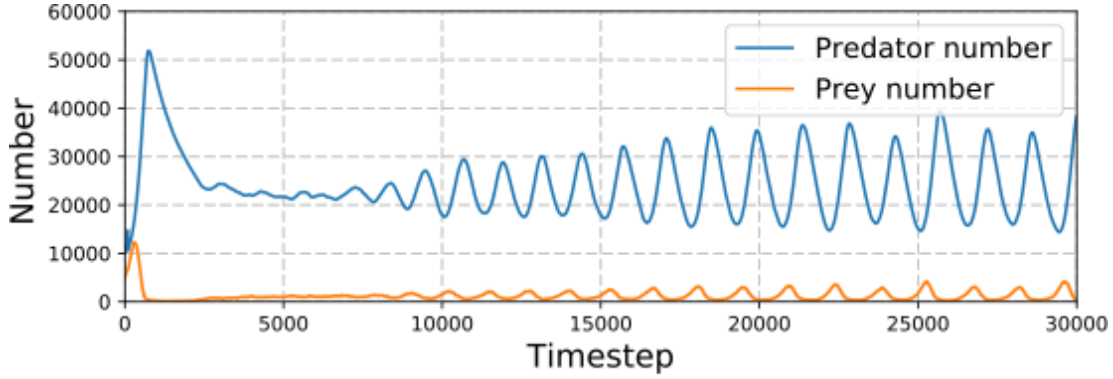


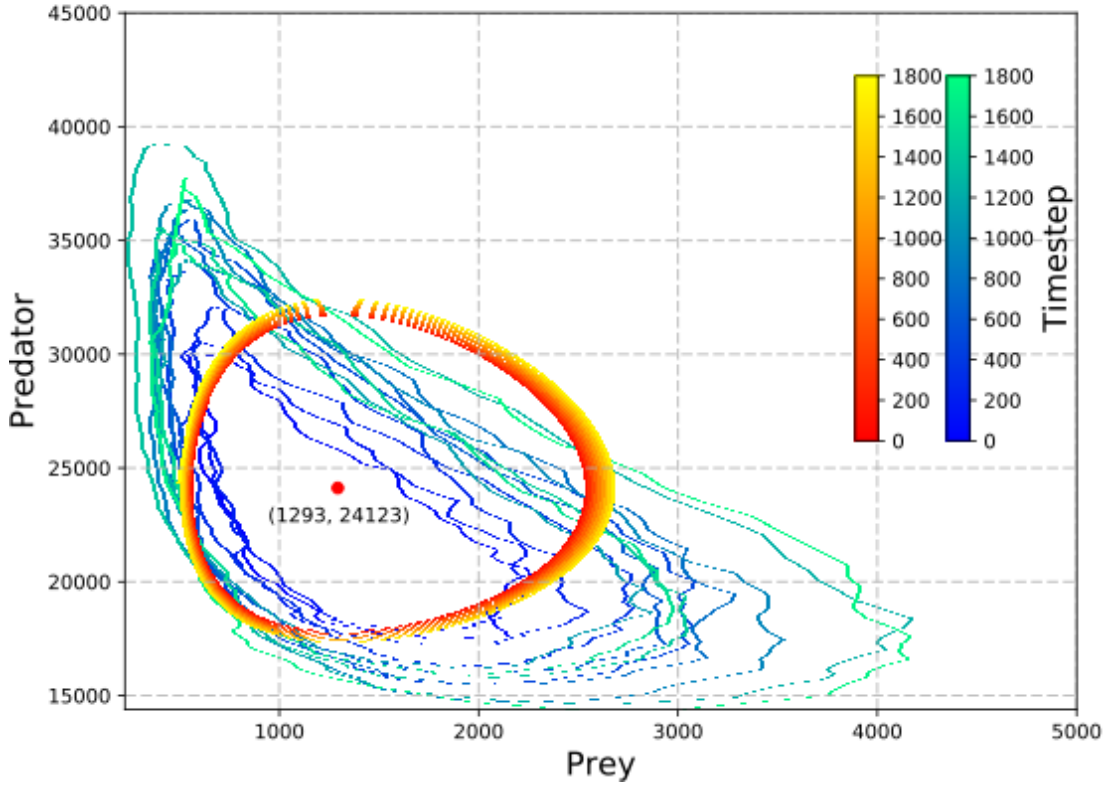
$$Q(s_t^i, a_t^i) \leftarrow Q(s_t^i, a_t^i) + \alpha[r_t^i + \gamma \max_{a' \in \mathcal{A}} Q(s_{t+1}^i, a') - Q(s_t^i, a_t^i)].$$

The action space A: {move forward, backward, left, right, rotate left, rotate right, stand still, join a group, and leave a group}.

Yaodong Yang , Lantao Yu , Yiwei Bai , Jun Wang , Weinan Zhang , Ying Wen , Yong Yu, , Dynamics of Artificial Populations by Million-agent Reinforcement Learning, 2017

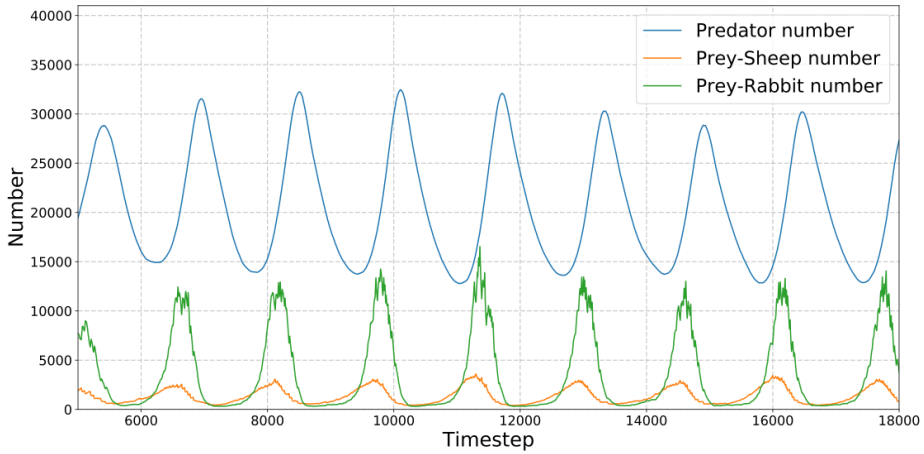
The Dynamics of the Artificial Population

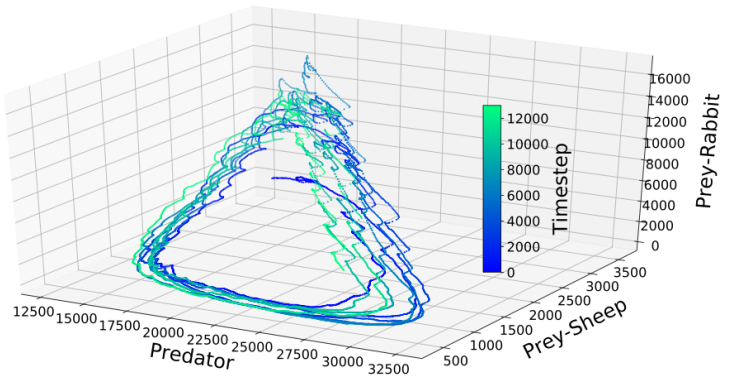


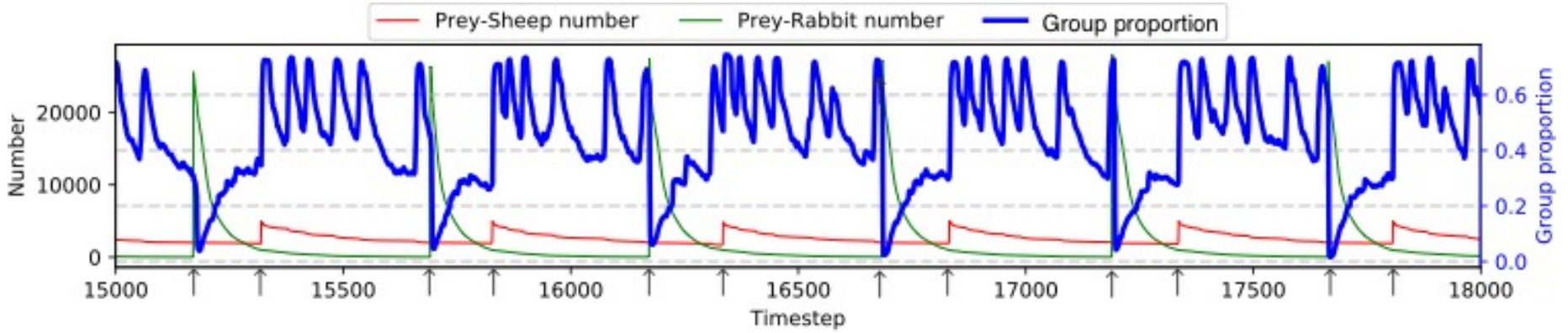


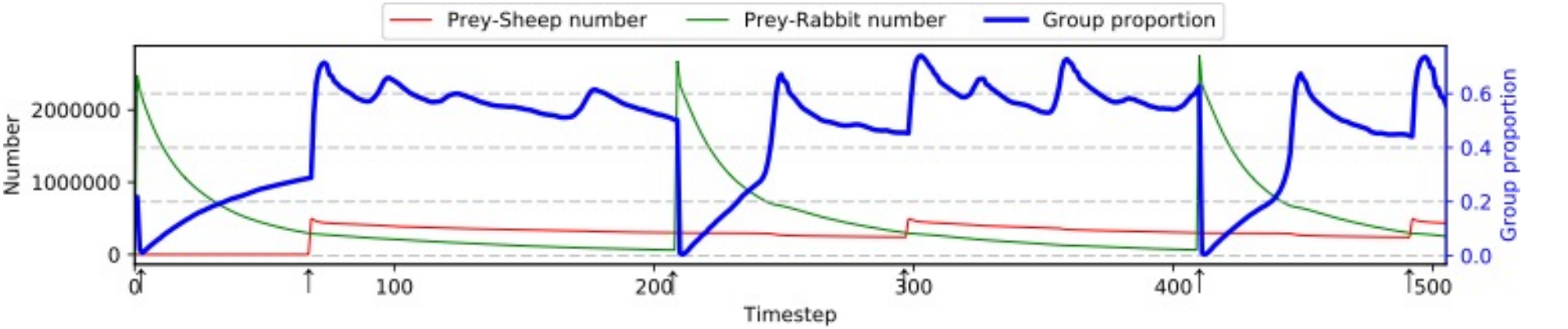
Yaodong Yang , Lantao Yu , Yiwei Bai , Jun Wang , Weinan Zhang , Ying Wen , Yong Yu, , Dynamics of Artificial Populations by Million-agent Reinforcement Learning, 2017

Tiger-sheep-rabbit: Grouping







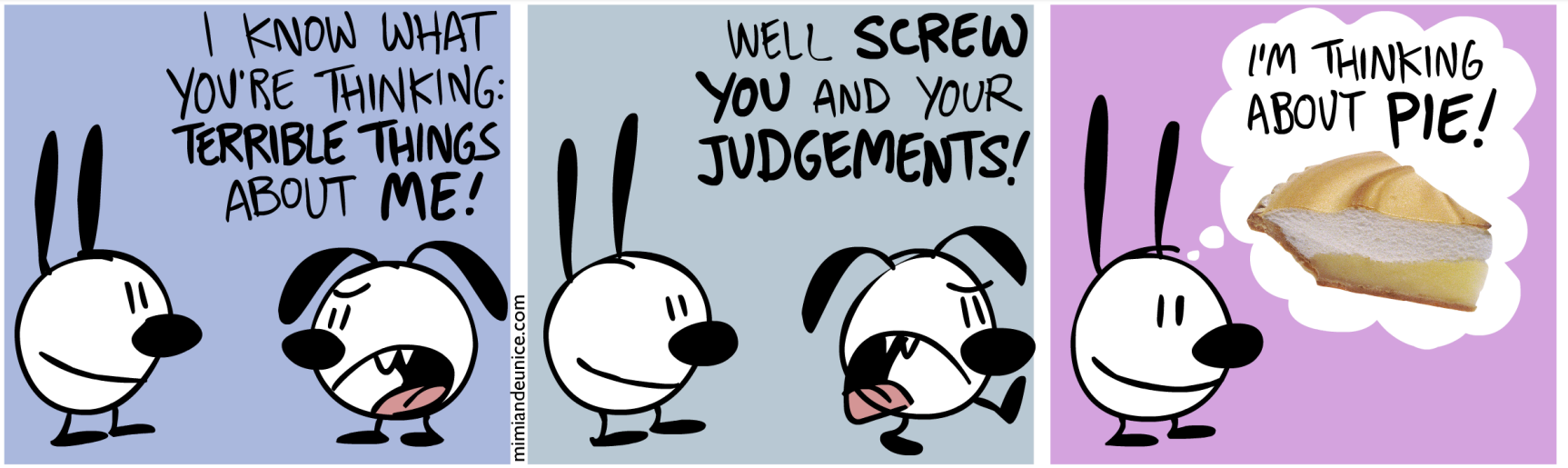


Yaodong Yang , Lantao Yu , Yiwei Bai , Jun Wang , Weinan Zhang , Ying Wen , Yong Yu, , Dynamics of Artificial Populations by Million-agent Reinforcement Learning, 2017

Content

- Emergent behaviours
- Agents modeling agents
- Learning communication
- Learning cooperation
- Many-agent learning

A theory of mind



Source: https://janetthomas.wordpress.com/2013/01/23/theory-of-mind/

Source: https://www.youtube.com/watch?v=2KVFMc7q2qs

Driving over Roundabout: Look others, assess, decide, act…

Gweon, H., and R. Saxe. "Developmental cognitive neuroscience of theory of mind." *Neural circuit development and function in the brain*. 2013. 367-377.

Regularized Opponent Model with Maximum Entropy Objective (ROMMEO)

- In cooperative multi-agent reinforcement learning, optimum is a strategy profile $(\pi^{1*}, \ldots, \pi^{n*})$ such that:

$$\mathbb{E}_{s\sim p_s, a_t^{i*}\sim\pi^{i*}, a_t^{-i*}\sim\pi^{-i*}}\left[\sum_{t=1}^{\infty}\gamma^t R^i(s_t, a_t^{i*}, a_t^{-i*})\right]$$

$$\geq \mathbb{E}_{s\sim p_s, a_t^{i}\sim\pi^{i}, a_t^{-i}\sim\pi^{-i}}\left[\sum_{t=1}^{\infty}\gamma^t R^i(s_t, a_t^{i}, a_t^{-i})\right]$$

$$\forall\pi\in\Pi, i\in(1\ldots n),\ R^i = R$$

where $\pi = (\pi^i, \pi^{-i})$ *and Agent* i*'s optimal policy is* π^{i*}.

Tian, Zheng, et al. "A regularized opponent model with maximum entropy objective." *AAAI* 2019.

Regularized Opponent Model with Maximum Entropy Objective (ROMMEO)

- In CMARL, a single agent's "optimality" depends on the joint actions (a_i , a_{-i})
  - Therefore, we define o_{it} = 1 only indicates that *agent* i*'s policy at time step* t *is optimal*.

$$P(o_t^i = 1 | o_t^{-i} = 1, s_t, a_t^i, a_t^{-i}) \propto \exp(R(s_t, a_t^i, a_t^{-i}))$$

- Therefore, we define agent i's objective as

$$\max \ \mathcal{J} \triangleq \log P(o_{1:T}^i = 1 | o_{1:T}^{-i} = 1)$$

  - If all agents play optimally, then agents can receive the maximum rewards, which is the optimum of the games

Tian, Zheng, et al. "A regularized opponent model with maximum entropy objective." *AAAI* 2019.

Regularized Opponent Model with Maximum Entropy Objective (ROMMEO)

- As we assume no knowledge of the optimal policies and the model of the environment, we treat them as latent variables:

$$
\begin{aligned}
&q(a^i_{1:T}, a^{-i}_{1:T}, s_{1:T} | o^i_{1:T} = 1, o^{-i}_{1:T} = 1) \\
&= P(s_1) \prod_t P(s_{t+1}|s_t, a_t) q(a^i_t | a^{-i}_t, s_t, o^i_t = o^{-i}_t = 1) \\
&\times q(a^{-i}_t | s_t, o^i_t = o^{-i}_t = 1) \\
&= P(s_1) \prod_t \underbrace{P(s_{t+1}|s_t, a_t)}_{\text{Transition}} \underbrace{\pi(a^i_t | s_t, a^{-i}_t)}_{\substack{\text{Own policy depending on} \\ \text{the action from} \\ \text{the opponent}}} \underbrace{\rho(a^{-i}_t | s_t)}_{\substack{\text{The opponent} \\ \text{model}}},
\end{aligned}
$$

Tian, Zheng, et al. “A regularized opponent model with maximum entropy objective.” *AAAI* 2019.

Regularized Opponent Model with Maximum Entropy Objective (ROMMEO)

- With the factorization, we derive a lower bound on the likelihood of optimality of agent *i*:

$$
\begin{aligned}
&\log P(o^i_{1:T} = 1 | o^{-i}_{1:T} = 1) \\
&\geq \mathcal{J}(\pi, \rho) \triangleq \sum_t \mathbb{E}_{(s_t, a^i_t, a^{-i}_t) \sim q} [R^i(s_t, a^i_t, a^{-i}_t) \\
&+ H(\pi(a^i_t | s_t, a^{-i}_t)) - D_{\mathrm{KL}}(\rho(a^{-i}_t | s_t) || P(a^{-i}_t | s_t))] \\
&= \sum_t \mathbb{E}_{s_t} [\underbrace{\mathbb{E}_{a^i_t \sim \pi, a^{-i}_t \sim \rho} [R^i(s_t, a^i_t, a^{-i}_t) + H(\pi(a^i_t | s_t, a^{-i}_t))]}_{\text{MEO}} \\
&\quad - \underbrace{\mathbb{E}_{a^{-i}_t \sim \rho} [D_{\mathrm{KL}}(\rho(a^{-i}_t | s_t) || P(a^{-i}_t | s_t))]}_{\text{Regularizer of } \rho}].
\end{aligned}
$$

Tian, Zheng, et al. “A regularized opponent model with maximum entropy objective.” *AAAI* 2019.

Recursive reasoning

Agent i: I believe that you believe

Camerer, Colin F., Teck-Hua Ho, and Juin-Kuan Chong. "A cognitive hierarchy model of games." *The Quarterly Journal of Economics* 119.3 (2004): 861-898.

Recursive reasoning

Agent i: I believe that you believe

Agent -i: I believe that you believe I believe…

Example: in the "beauty contest" game, in which players are asked to pick numbers from 0 to 100, and the player whose number is closest to 2/3 of the average wins a prize.

Camerer, Colin F., Teck-Hua Ho, and Juin-Kuan Chong. "A cognitive hierarchy model of games." *The Quarterly Journal of Economics* 119.3 (2004): 861-898.

Recursive reasoning

Agent i: I believe that you believe

Agent -i: I believe that you believe that I believe…

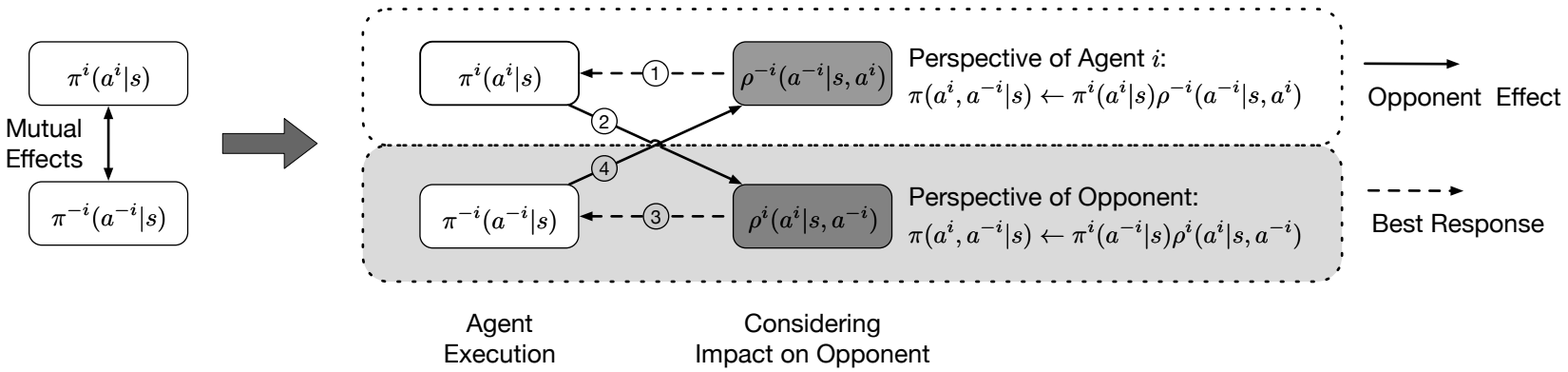
Example: in the "beauty contest" game, in which players are asked to pick numbers from 0 to 100, and the player whose number is closest to 2/3 of the average wins a prize.

Results: the group average is typically between 20 and 35

Camerer, Colin F., Teck-Hua Ho, and Juin-Kuan Chong. "A cognitive hierarchy model of games." *The Quarterly Journal of Economics* 119.3 (2004): 861-898.

Probabilistic recursive reasoning

$$\pi_\theta(a^i, a^{-i}|s) = \underbrace{\pi^i_{\theta^i}(a^i|s)\pi^{-i}_{\theta^{-i}}(a^{-i}|s, a^i)}_{\text{Agent } i\text{'s perspective}} = \underbrace{\pi^{-i}_{\theta^{-i}}(a^{-i}|s)\pi^i_{\theta^i}(a^i|s, a^{-i})}_{\text{The opponents' perspective}}.$$



① agent *i* takes the best response after considering all the potential consequences of opponents' actions given its own action a_i.
② how agent *i* behaves in the environment serves as the prior for the opponents to learn how their actions would affect a_i.
③ similar to ① , opponents take the best response to agent *i*.
④ similar to ② , opponents' actions are the prior knowledge to agent *i* on estimating how a_i will affect the opponents. Looping from step 1 to 4 forms recursive reasoning.

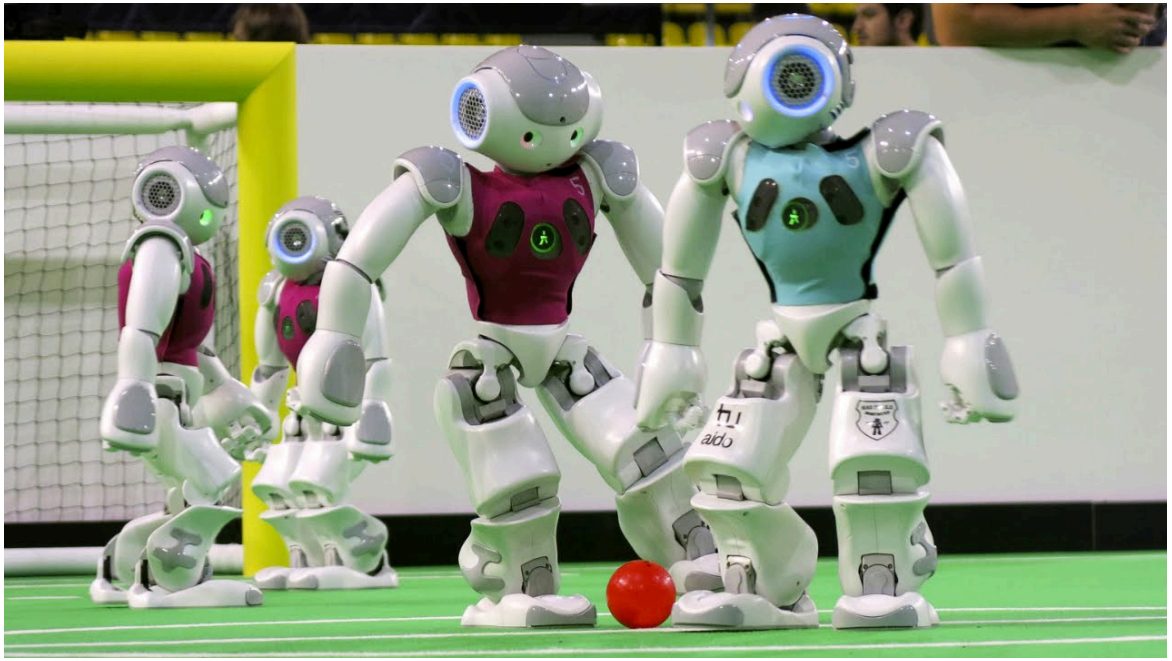
Wen Ying et al, Probabilistic Recursive Reasoning for Multi-Agent Reinforcement Learning, 2018

Content

- Emergent behaviours
- Agents modeling agents
- Learning communication
- Learning cooperation
- Many-agent learning

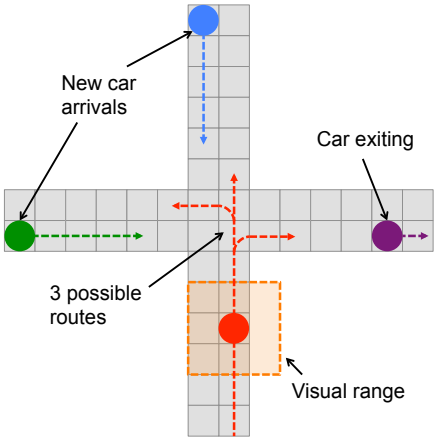
Communications among agents

- AI require the collaboration of multiple agents
- the communication between agents is vital to coordinate the behaviour of each individual

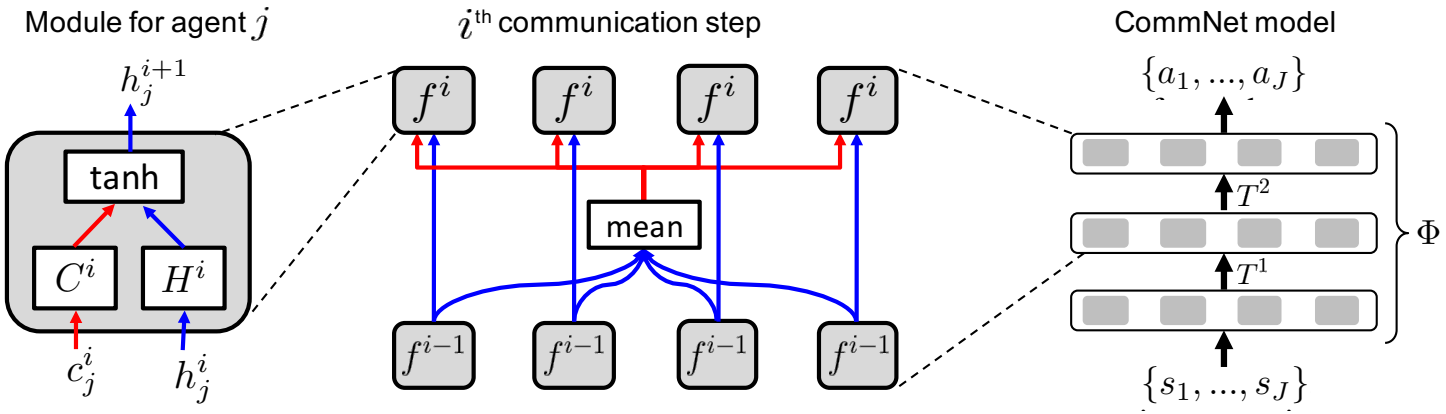


CommNets

- Full cooperation between agents
- The model consists of multiple agents and the communication between them is learned alongside their policy.



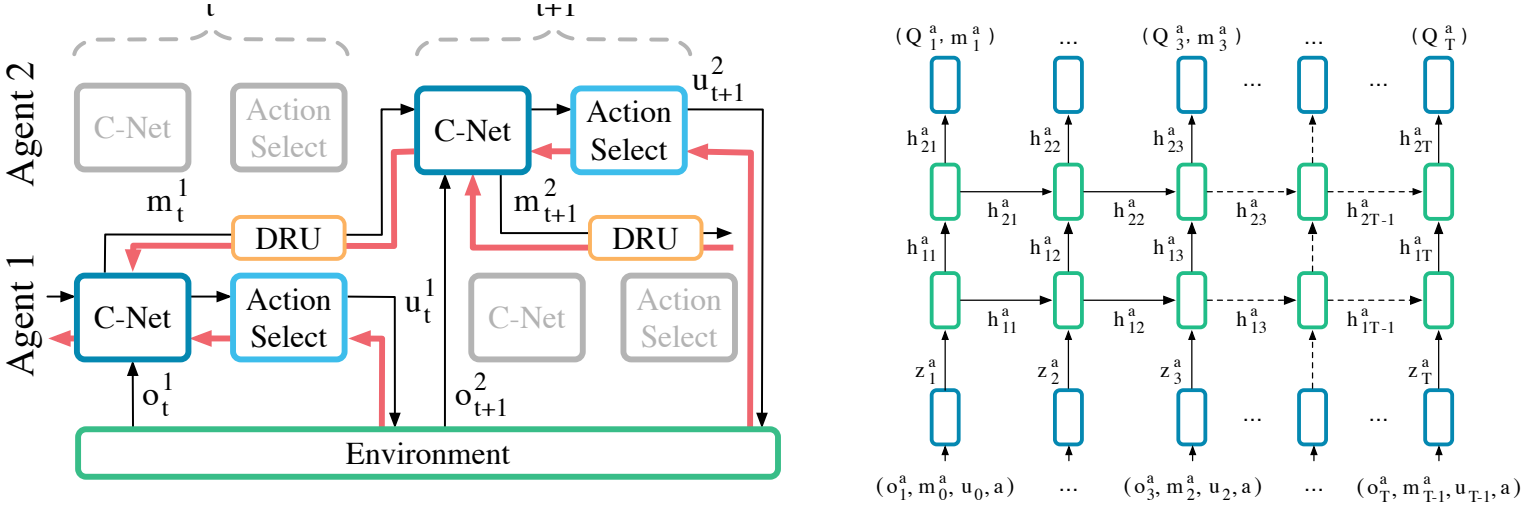
pass the through the junction without colliding



Sukhbaatar, Sainbayar, and Rob Fergus. "Learning multiagent communication with backpropagation." *Advances in Neural Information Processing Systems*. 2016.

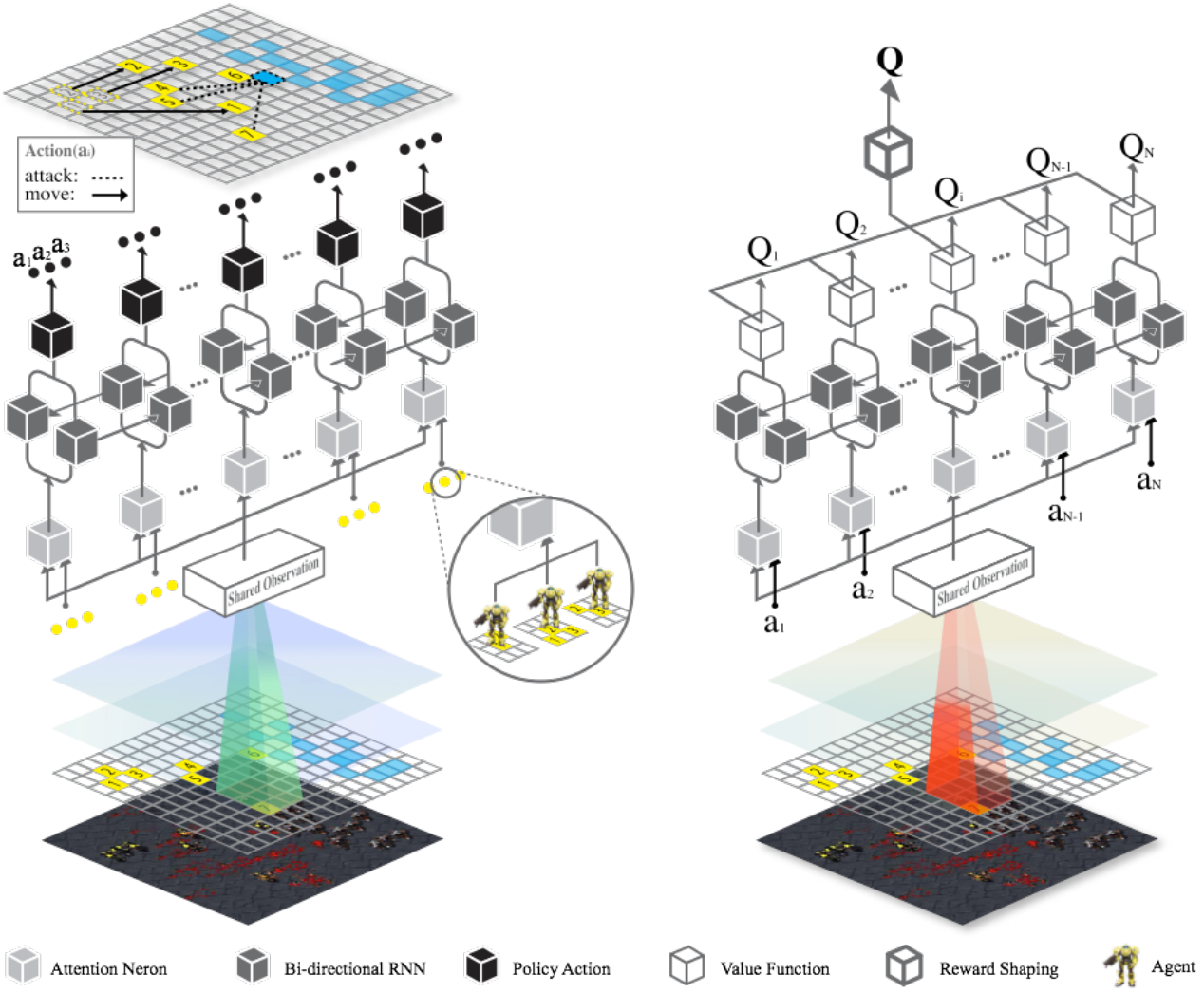
Differentiable Inter-Agent Learning (DIAL)

- Uses centralized learning but decentralised execution
  - during learning, agents can backpropagate error derivatives through (noisy) communication channels



Foerster J, Assael IA, de Freitas N, Whiteson S. Learning to communicate with deep multi-agent reinforcement learning. InAdvances in Neural Information Processing Systems 2016 (pp. 2137-2145).

Bidirectional-Coordinated nets (BiCNet)



(a) Multiagent policy networks with grouping (b) Multiagent Q networks with reward shaping

Peng Peng, Quan Yuan, Ying Wen, Yaodong Yang, Zhenkun Tang, Haitao Long, Jun Wang, Multiagent Bidirectionally-Coordinated Nets for Learning to Play StarCraft Combat Games, 2017

Content

- Emergent behaviours
- Agents modeling agents
- Learning communication
- Learning cooperation
- Many-agent learning

Deterministic Policy Gradient

- Deterministic policy gradient (DPG) can hand continuous action spaces
- DPG typically is based on actor critic:
  - The critic estimates the action-value function, which is then maximized by the actor

Critic: $$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} \mathbb{E}_{s,a,r,s'}\left[\left(y - Q(s,a|\boldsymbol{\phi})\right)^2\right]$$

where $$y = \mathbb{E}_{s'}\left[r + \gamma \max_{a' \in \mathcal{A}} Q(s', a'|\boldsymbol{\phi}')\right]$$

After optimizing the critic, the actor is learned by maximizing the expected Q-value

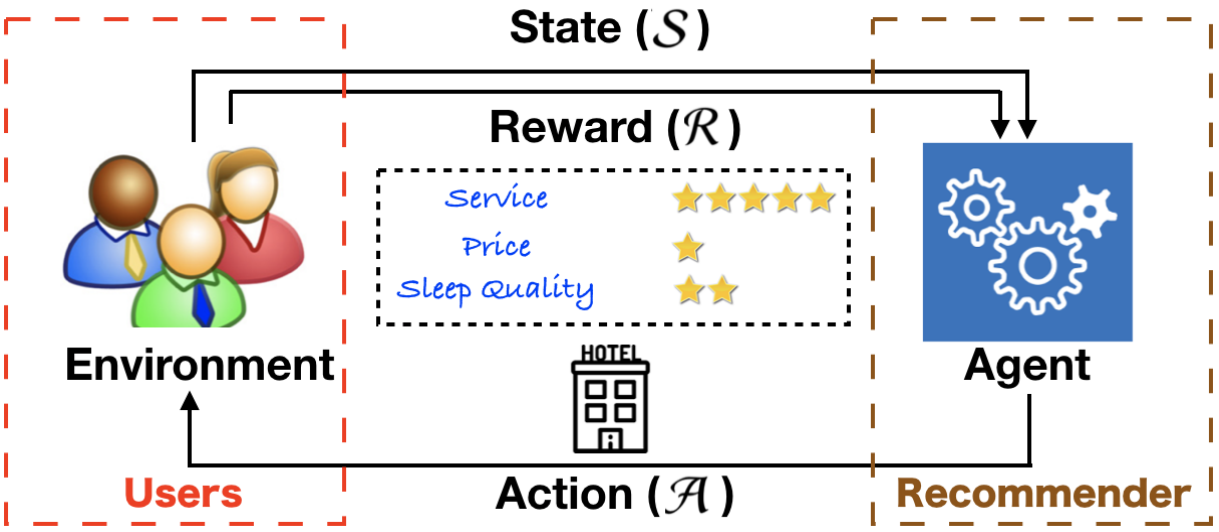
Actor: $$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{s'}\left[Q(s', \mu(s'|\boldsymbol{\theta})|\boldsymbol{\phi})\right]$$

where the deterministic policy is $a' = \mu(s'|\boldsymbol{\theta})$

Silver, David, et al. "Deterministic policy gradient algorithms." 2014.

Multi-Objective Optimization

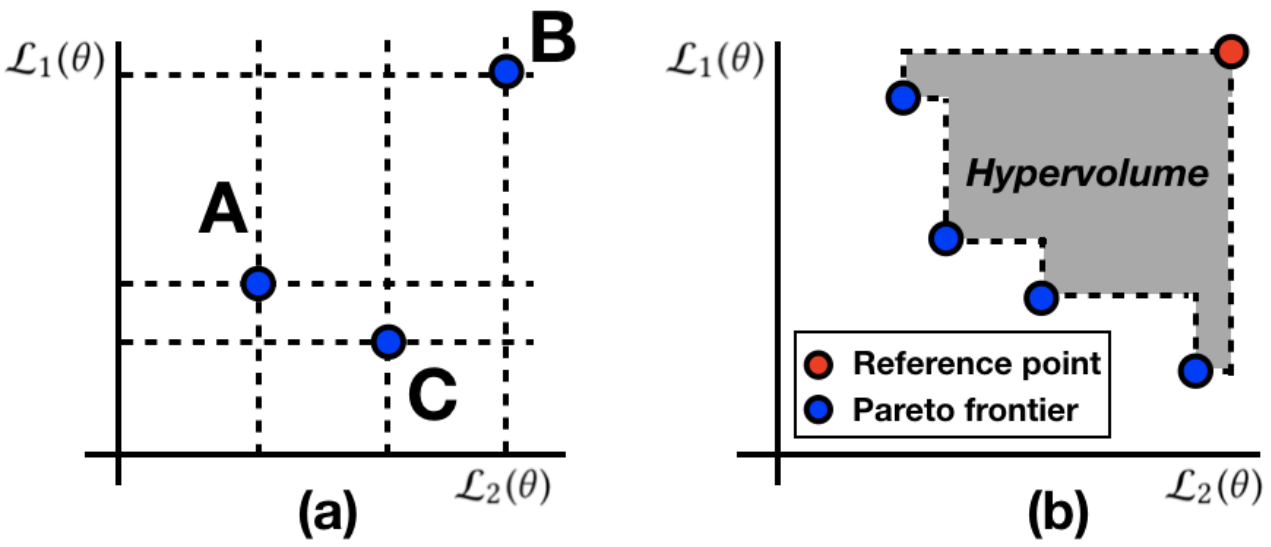
- Multi-objective MDP problem: recommenders
  - The agent selects an action (recommends an item) based on the current state, and
  - the environment generate the next state and the reward reflecting the user preference, which is usually multi-dimensional in practice.



Multi-Objective Control with Deterministic Policy Gradient: A Case Study of Recommender Systems 2020

Multi-Objective Optimization

- Pareto optimization is a common strategy to solve multi-objective problems
  - Suppose we need to minimize a series of loss functions $L(\theta) = \{L_1(\theta), L_2(\theta), ..., L_M(\theta)\}$.
  - The optimal θ's for different L_i's are various

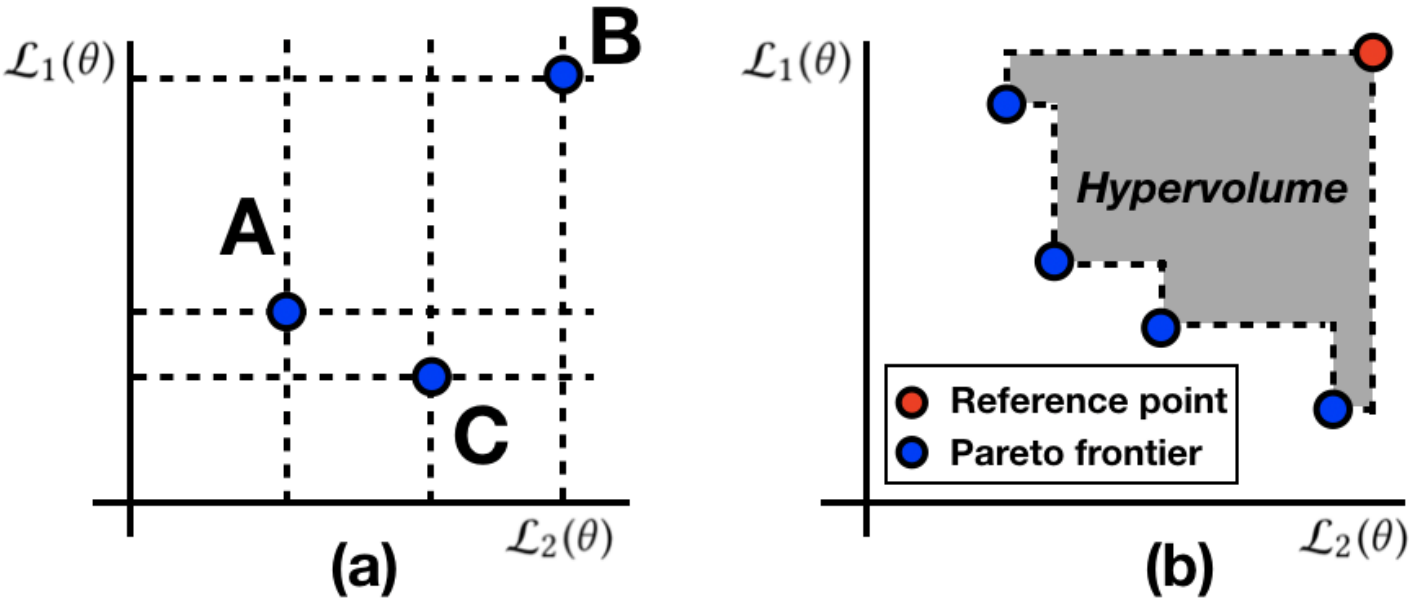


Multi-Objective Control with Deterministic Policy Gradient: A Case Study of Recommender Systems 2020

Multi-Objective Optimization

- Pareto dominance: For two parameters θ_A and θ_B in domain Ω, θ_A is said to dominant θ_B ($\theta_A \prec \theta_B$) if and only if $L_i(\theta_A) \leq L_i(\theta_B)$, $\forall i \in \{1,2,...M\}$ and $L_i(\theta_A) < L_i(\theta_B)$, $\exists i \in \{1,2,...M\}$.

The parameters of point A and C can dominant that of point B, while there is no dominance relationship between A and C



Multi-Objective Control with Deterministic Policy Gradient: A Case Study of Recommender Systems 2020

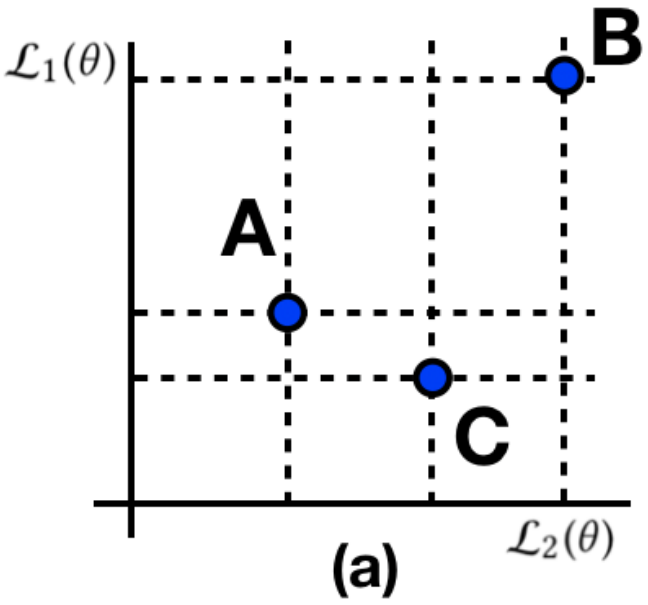
Multi-Objective Optimization

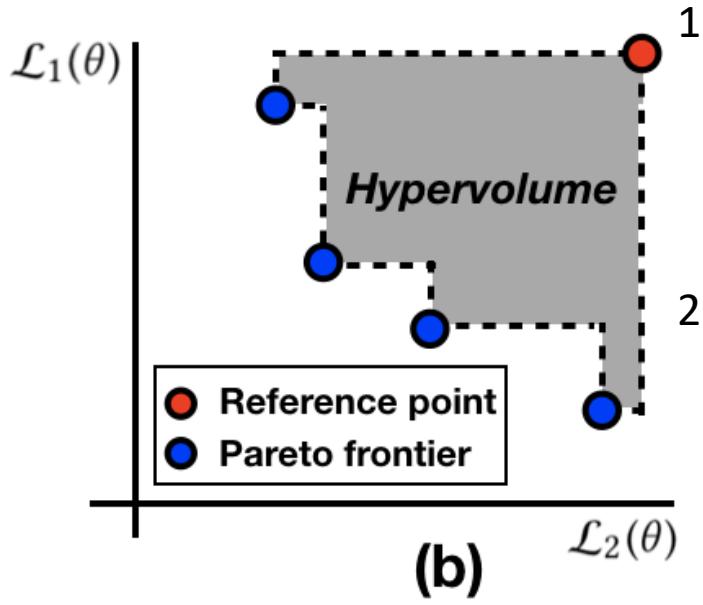
- The goal of Pareto optimization is to learn a parameter θ^*, such that it cannot be further improved to lower one loss function without increasing the other ones
  - The parameter θ^* is called Pareto efficiency
- Pareto efficiency (optimality): For a parameter θ^*, if there is no parameter $\hat{\theta} \in \Omega$, such that $\hat{\theta} \prec \theta^*$. Then we say θ^* is a Pareto efficiency or Pareto optimality solution.

Multi-Objective Control with Deterministic Policy Gradient: A Case Study of Recommender Systems 2020

Multi-Objective Optimization

- There can be multiple Pareto efficiency parameters, and all these solutions form the Pareto set, and the image (the value of the objective functions) of the Pareto set is defined as the Pareto frontier.





Evaluation:

1. Set up a reference point, which is dominated by all the Pareto efficiency points.
2. The value of hypervolume is the area bounded by the approximated front and the reference point

Multi-Objective Control with Deterministic Policy Gradient: A Case Study of Recommender Systems 2020

Multi-Objective Deterministic Policy Gradient

- Multi-objective Critic
  - suppose our system is composed of M objectives, then the critics are defined as:

    $\{Q_1(s, a|\boldsymbol{\phi}_1), Q_2(s, a|\boldsymbol{\phi}_2), \ldots, Q_M(s, a|\boldsymbol{\phi}_M)\}$
  - where the critics are optimized by minimizing:

$$\sum_{i=1}^{Z}(y_{i,m} - Q_m(s_i, a_i|\boldsymbol{\phi}_m))^2, \ \forall m = 1, 2, \ldots M$$

  - where Z is the number of samples, and for each instance i, $y_{i,m} = r_{i,m} + \gamma Q_m(s_{i+1}, a_{i+1}|\boldsymbol{\phi}'_m)$. $r_{i,m}$ is the reward for the mth objective

Multi-Objective Control with Deterministic Policy Gradient: A Case Study of Recommender Systems 2020

Multi-Objective Deterministic Policy Gradient

- Pareto-optimal Actor
  - As we have multiple Q functions, our actor is learned to find a Pareto efficiency solution for the multiple objectives:

$$\mathrm{L}(\boldsymbol{\theta}) = \{\mathrm{L}_1(\boldsymbol{\theta}), \mathrm{L}_2(\boldsymbol{\theta}), \ldots, \mathrm{L}_M(\boldsymbol{\theta})\},$$

$$\text{where } \mathrm{L}_m(\boldsymbol{\theta}) = -\mathrm{E}_s[Q_m(s, \mu(s|\boldsymbol{\theta}))].$$

- Gradient- based Pareto optimization:
  - We define a weighted objective:

$$l(\theta) = -\sum_{m=1}^{M} w_m \mathbb{E}_s[Q_m(s, \mu(s|\theta))] \quad \boldsymbol{w} = \{w_1, w_2, \ldots, w_M\}$$

where weights are continually changed in the learning process.

Multi-Objective Control with Deterministic Policy Gradient: A Case Study of Recommender Systems 2020

Multi-Objective Deterministic Policy Gradient

- Gradient- based Pareto optimization:
  - First, we define a weighted objective:

$$l(\boldsymbol{\theta}) = -\sum_{m=1}^{M} w_m \mathbb{E}_s[Q_m(s, \mu(s|\boldsymbol{\theta}))] \qquad \boldsymbol{w} = \{w_1, w_2, ..., w_M\}$$

  - We then obtain the weights by solving the following quadratic programming problem

$$\min_{\boldsymbol{w}} || \sum_{m=1}^{M} w_m \nabla_{\boldsymbol{\theta}} \mathbb{E}_s[Q_m(s, \mu(s|\boldsymbol{\theta}))] ||_2^2$$

$$s.t. \sum_{m=1}^{M} e_{k,m} w_m \geq b_k, \forall\, k \in [1, K]$$

$$\sum_{m=1}^{M} w_m = 1,\ w_m \geq 0, \forall\, m \in [1, M]$$

Several pre-defined preference vectors $\{\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_K\}$ and values $\{b_1, b_2, ..., b_K\}$ are introduced to constrain $\boldsymbol{w}$, where $\boldsymbol{e}_k = \{e_{k,1}, e_{k,2}, ..., e_{k,M}\}$, $\sum_{m=1}^{M} e_{k,m} = 1$, $e_{k,m} \geq 0$, and $b_k \geq 0$

For example, if $\boldsymbol{e}_k$ is an one-hot vector, the constraint aims to set an importance-level for the corresponding objective by b_k

Multi-Objective Control with Deterministic Policy Gradient: A Case Study of Recommender Systems 2020

Multi-Objective Deterministic Policy Gradient

- Theorem: If $\boldsymbol{w}$ is determined by solving the quadratic programming (QP) problem, then either one of the following holds:
  - The objective of the optimization problem is 0, then the local Pareto optimality is achieved.
  - $\boldsymbol{d} = \sum_{m=1}^{M} w_m \nabla_{\boldsymbol{\theta}} \mathbb{E}_s [Q_m(s, \mu(s|\boldsymbol{\theta}))]$ is a gradient direction which can lower all the objectives in L($\boldsymbol{\theta}$)

Multi-Objective Deterministic Policy Gradient

- We can write the Lagrangian of the problem:

$$|| \sum_{m=1}^{M} w_m \nabla_{\boldsymbol{\theta}} \mathbb{E}_s [Q_m(s, \mu(s|\boldsymbol{\theta}))]||_2^2$$

with $\lambda_k \geq 0, \beta \geq 0, \forall k \in [1, K]$.

$$+ \sum_{k=1}^{K} \lambda_k (b_k - \sum_{m=1}^{M} e_{k,m} w_m) + \beta (1 - \sum_{m=1}^{M} w_m)$$

- The KTT condition gives:

$$(\sum_{m=1}^{M} w_m \nabla_{\boldsymbol{\theta}} \mathbb{E}_s [Q_m(s, \mu(s|\boldsymbol{\theta}))])^T \nabla_{\boldsymbol{\theta}} \mathbb{E}_s [Q_m(s, \mu(s|\boldsymbol{\theta}))]$$

$$= \sum_{k=1}^{K} \lambda_k e_{k,m} + \beta \geq 0, \forall m \in [1, M]$$

Recall that $\boldsymbol{d} = \sum_{m=1}^{M} w_m \nabla_{\boldsymbol{\theta}} \mathbb{E}_s [Q_m(s, \mu(s|\boldsymbol{\theta}))]$ thus, for all m, we have $\boldsymbol{d}^T \nabla_{\boldsymbol{\theta}} \mathbb{E}_s [Q_m(s, \mu(s|\boldsymbol{\theta}))] \geq 0$ which means $\boldsymbol{d}$ is a direction that can increase all $\mathbb{E}s[Q_m(s,\mu(s|\boldsymbol{\theta}))], \forall m \in [1,M]$.

Multi-Objective Control with Deterministic Policy Gradient: A Case Study of Recommender Systems 2020

Multi-agent deep deterministic policy gradient (MADDPG)

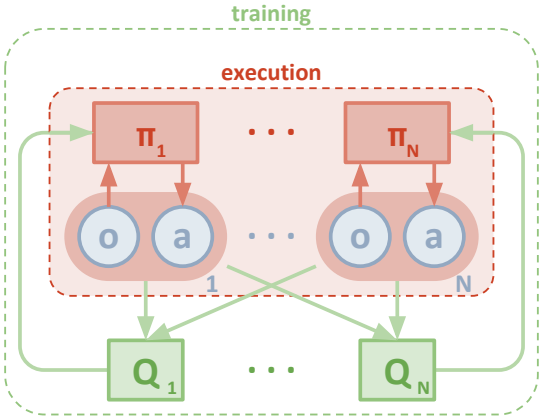
- If consider *N* continuous policies μ_i w.r.t. parameters θ_i, the gradient can be written as

$$\nabla_{\theta_i} J(\boldsymbol{\mu}_i) = \mathbb{E}_{\mathbf{x}, a \sim \mathcal{D}}[\nabla_{\theta_i} \boldsymbol{\mu}_i(a_i|o_i) \nabla_{a_i} Q_i^{\boldsymbol{\mu}}(\mathbf{x}, a_1, ..., a_N)|_{a_i = \boldsymbol{\mu}_i(o_i)}]$$

where D contains the tuples (x, x′, a_1, . . . , a_N, r_1, . . . , r_N), experience replay of all agents

- The Q-function can be optimized by loss function:

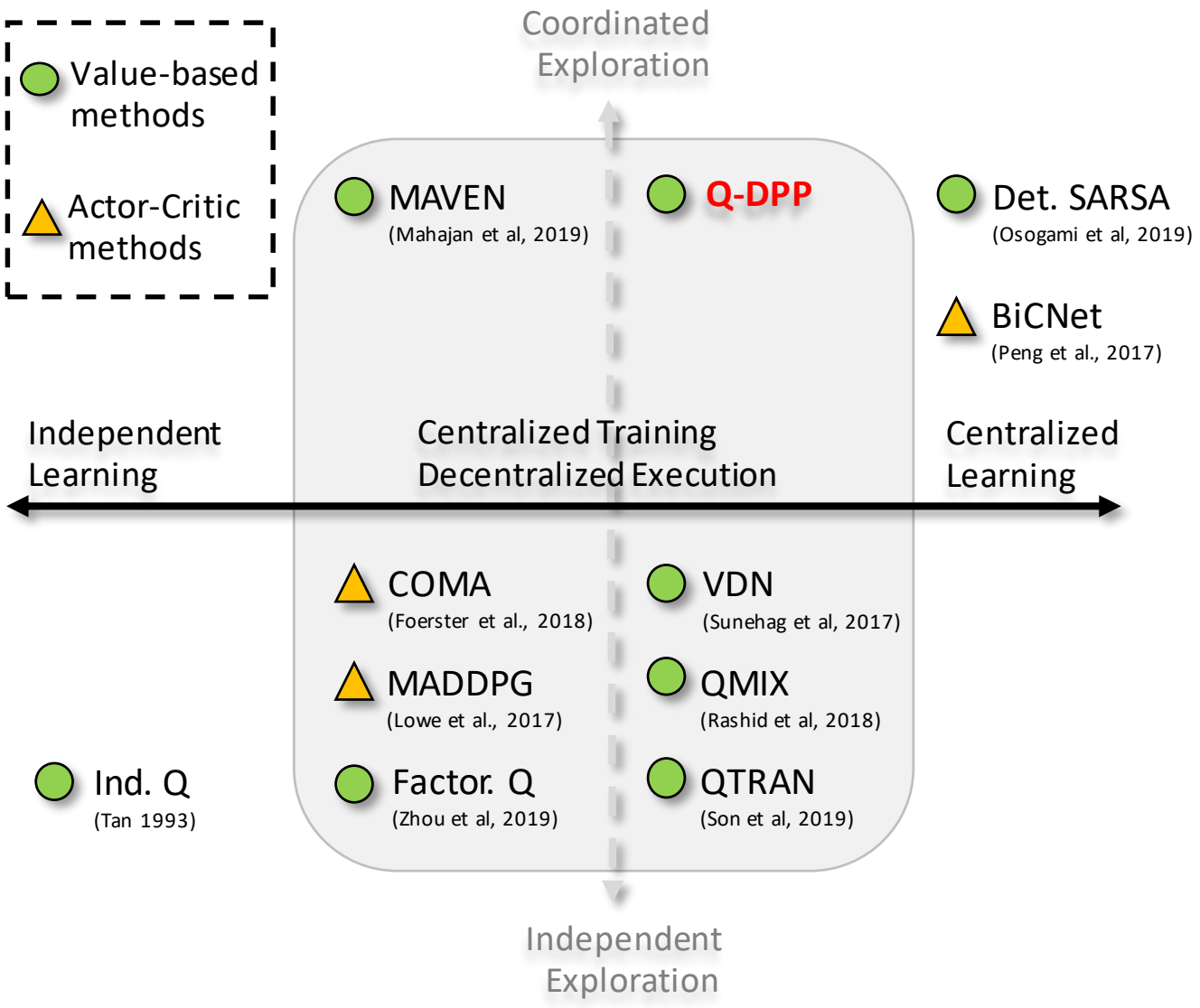
$$\mathcal{L}(\varphi_i) = \mathbb{E}_{\mathbf{x}, a, r, \mathbf{x}'}[(Q_i^{\boldsymbol{\mu}}(\mathbf{x}, a_1, \ldots, a_N) - y)^2], \quad y = r_i + \gamma\, Q_i^{\boldsymbol{\mu}'}(\mathbf{x}', a'_1, \ldots, a'_N)\big|_{a'_j = \boldsymbol{\mu}'_j(o_j)}$$



Centralized training and decentralized execution

Lowe, Ryan, et al. “Multi-agent actor-critic for mixed cooperative-competitive environments.” *Advances in neural information processing systems*. 2017.

Multiagent Collaboration

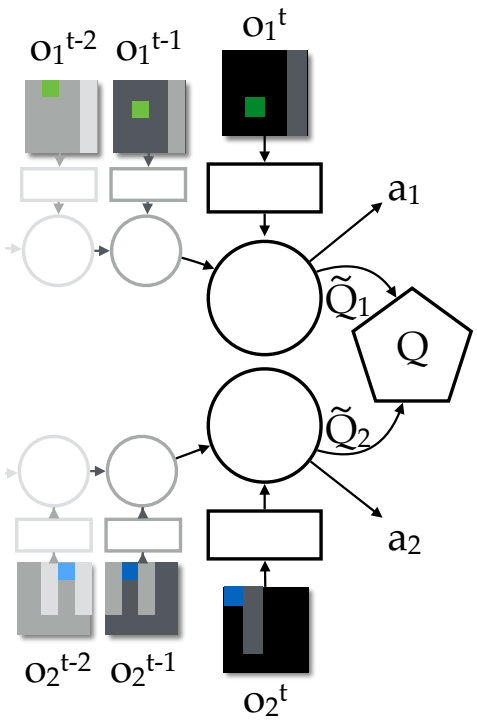


Value-decomposition networks (VDN)

- Cooperative multi-agent RL shares a single joint reward signal
  - Need to learn to decompose the team value function into agent-wise value functions

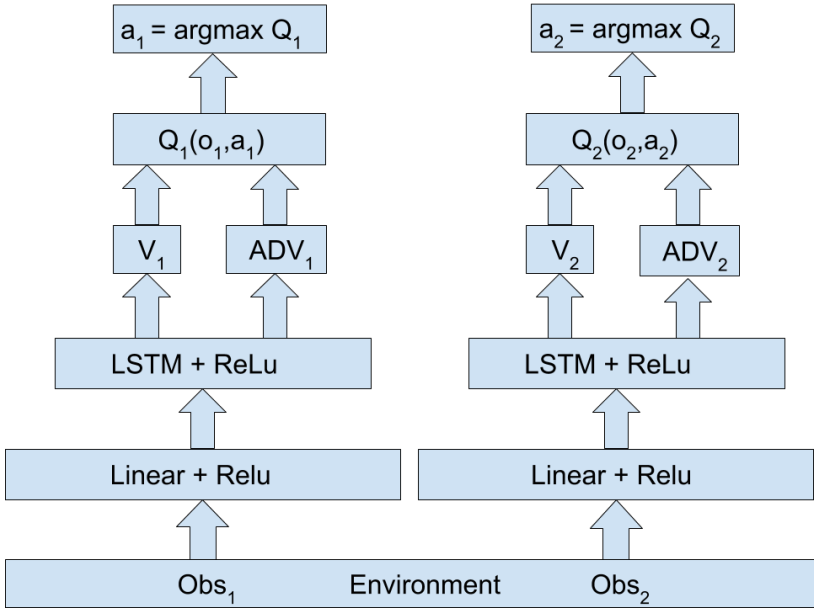
$$Q((h^1, h^2, ..., h^d), (a^1, a^2, ..., a^d)) \approx \sum_{i=1}^{d} \tilde{Q}_i(h^i, a^i)$$

  - $\tilde{Q}_i$ is learned implicitly rather than from any reward specific to agent I
  - h^i is from the recurrent layer over time
- Although learning requires some centralization, the learned agents can be deployed independently

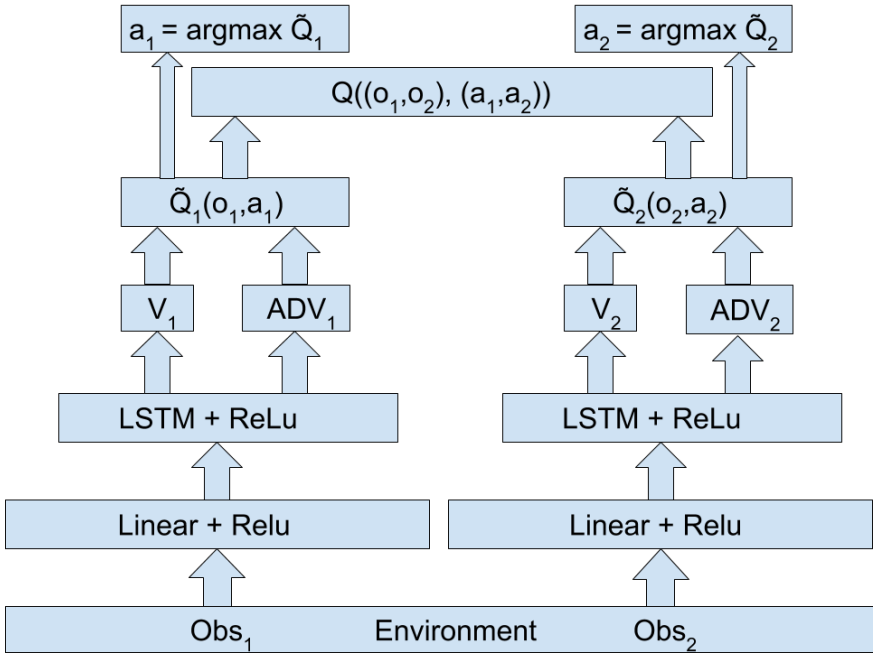


Sunehag, Peter, et al. "Value-decomposition networks for cooperative multi-agent learning." *arXiv preprint arXiv:1706.05296* (2017).

Value-decomposition networks (VDN)



Independent Agents Architecture



Value-Decomposition Individual Architecture

Sunehag, Peter, et al. "Value-decomposition networks for cooperative multi-agent learning." *arXiv preprint arXiv:1706.05296* (2017).

Multiagent Collaboration

Definition 2 (Decentralizable Cooperative Tasks, a.k.a. Individual-Global-Max Condition (Son et al., 2019)). *A cooperative task is decentralizable if* $\exists\{Q_i\}_{i=1}^N$ *such that* $\forall \boldsymbol{\tau} \in \tau^N, \boldsymbol{a} \in \mathcal{A}^N$,

$$\arg\max_{\boldsymbol{a}} Q^{\boldsymbol{\pi}}(\boldsymbol{\tau}, \boldsymbol{a}) = \begin{bmatrix} \arg\max_{a_1} Q_1(\tau_1, a_1) \\ \vdots \\ \arg\max_{a_N} Q_N(\tau_N, a_N) \end{bmatrix}. \quad (3)$$

- This suggests that local maxima on the extracted value function per every agent needs to amount to the global maximum on the joint value function.
- A key challenge for CTDE (Centralized Training Decentralized Execution) methods is, then, how to correctly extract each of the agent's individual Q-function $\{Q_i\}$, and as such an executable policy, from a centralized Q-function Q^{π}.

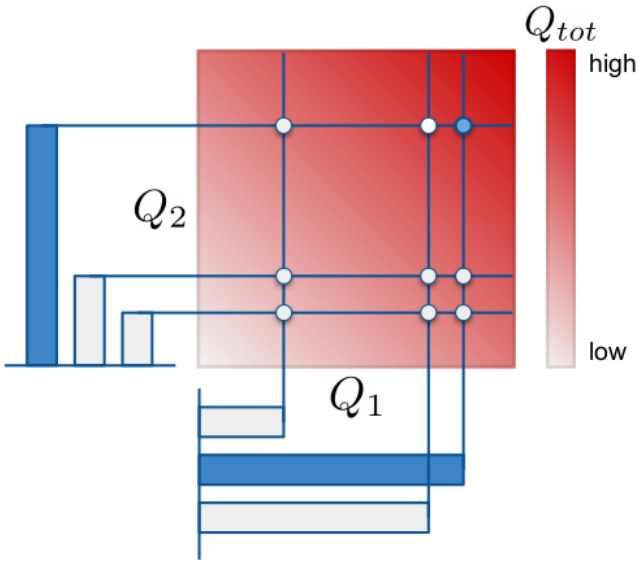
Yaodong Yang et al., Multi-Agent Determinantal Q-Learning, 2020

QMIX: Monotonic Value Function Factorization

- Unlike VDN, QMIX reinforces that a global argmax performed on Q_{tot} yields the same result as a set of individual argmax operations performed on each Q_a

$$\frac{\partial Q_{tot}}{\partial Q_a} \geq 0, \ \forall a.$$

- The constraint can be satisfied due to the monotonicity of Q_{tot}.

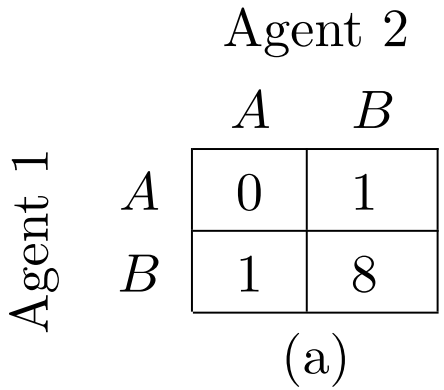


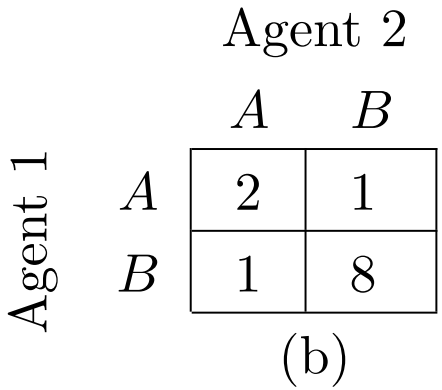
Per-agent action-value scores Q_a are fed into the monotonic function $Q_{tot}(Q_1, Q_2)$. The maximum Q_a for each agent is shown in blue, which corresponds to the maximum Q_{tot} also shown in blue

Rashid, Tabish, et al. "Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning." *arXiv preprint arXiv:2003.08839* (2020).

QMIX: Monotonic Value Function Factorization

- However, the monotonic constraint prevents QMIX from representing any value function
  - Intuitively, any value function for which an agent's best action depends on the actions of the other agents at the same time step will not factorise appropriately, and
  - hence cannot be perfectly represented by QMIX.





(a) A monotonic payoff matrix, (b) a non-monotonic payoff matrix

Rashid, Tabish, et al. "Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning." *arXiv preprint arXiv:2003.08839* (2020).

Exploratory Action Noise

- Agents in the system provide a constantly changing background in which each agent needs to learn its task
  - As a consequence, agents need to extract the underlying reward signal from the noise of other agents acting within the environment
- This learning noise can have a significant impact on the resultant system performance

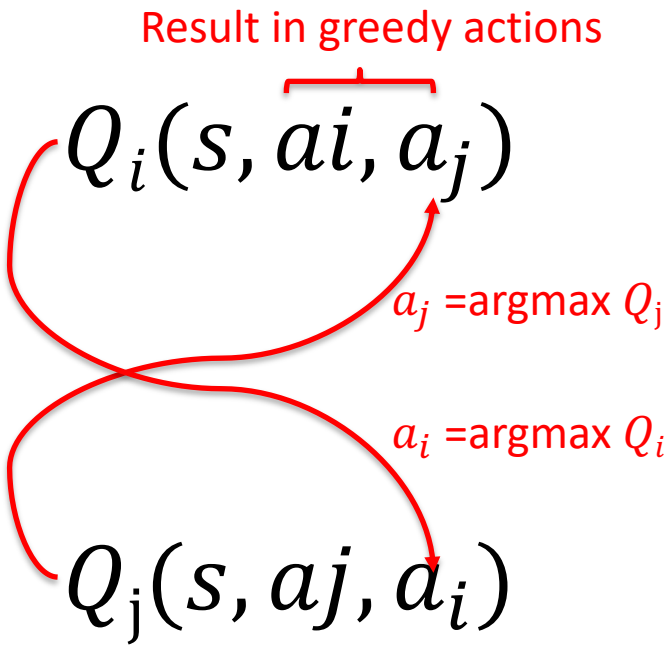
$$Q_i(s, ai, a_j)$$

Condition on other agent actions: a_j but they are also exploring – the actual a_j contains some element of exploration and not their intended actions

C. Holmesparker, M. E. Taylor, A. K. Agogino, and K. Tumer. Clean rewards to improve coordination by removing exploratory action noise. In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 03, pages 127–134, 2014.

CLEAN rewards

- **Coordinated** Learning without **Exploratory Action Noise** (CLEAN) aims to remove exploratory noise present in the global reward
  - This is achieved by private exploration
- Specifically, at each learning episode, each agent executes an action by following its greedy policy (i.e. without exploration);
- then all the agents receive a global re-ward.
- Each agent then privately computes the (global) reward it would have received had it executed an exploratory action, while the rest of the agents followed their greedy policies.



C. Holmesparker, M. E. Taylor, A. K. Agogino, and K. Tumer. Clean rewards to improve coordination by removing exploratory action noise. In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 03, pages 127–134, 2014.

CLEAN rewards

- CLEAN rewards were defined:

$$D_i = \widehat{R}_i(s, a_i^c, aj) - Ri\left(s, a_i\ , aj\right)$$

  - where $(a_i\ , aj)$ is the joint action executed when all agents followed their greedy policies,
  - a_i^c is the counterfactual (offline) action taken by agent i following ε-greedy,
  - R_i is the reward of agent i received when all agents executed their greedy policies and
  - $\widehat{R}_i(s, a_i^c, aj)$ is the counterfactual (offline) reward agent i would have received, had it executed the counterfactual action a_i^c, instead of action a_i, while the rest of the agents followed their greedy policies.
- Each agent then uses the following formula to update its Q-values:

$$Q_i(s, a_i^c, aj) \leftarrow Q_i(s, a_i^c, aj) + \alpha(D_i - Q_i(s, a_i^c, aj))$$

- which removes the exploratory noise caused by other agents and
- allow each agent to effectively determine which actions are beneficial or not

C. Holmesparker, M. E. Taylor, A. K. Agogino, and K. Tumer. Clean rewards to improve coordination by removing exploratory action noise. In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 03, pages 127–134, 2014.

CLEAN rewards: Experiment

- The Gaussian Squeeze Domain (GSD):
  - There is a set of agents in which each agent contributes to a system objective

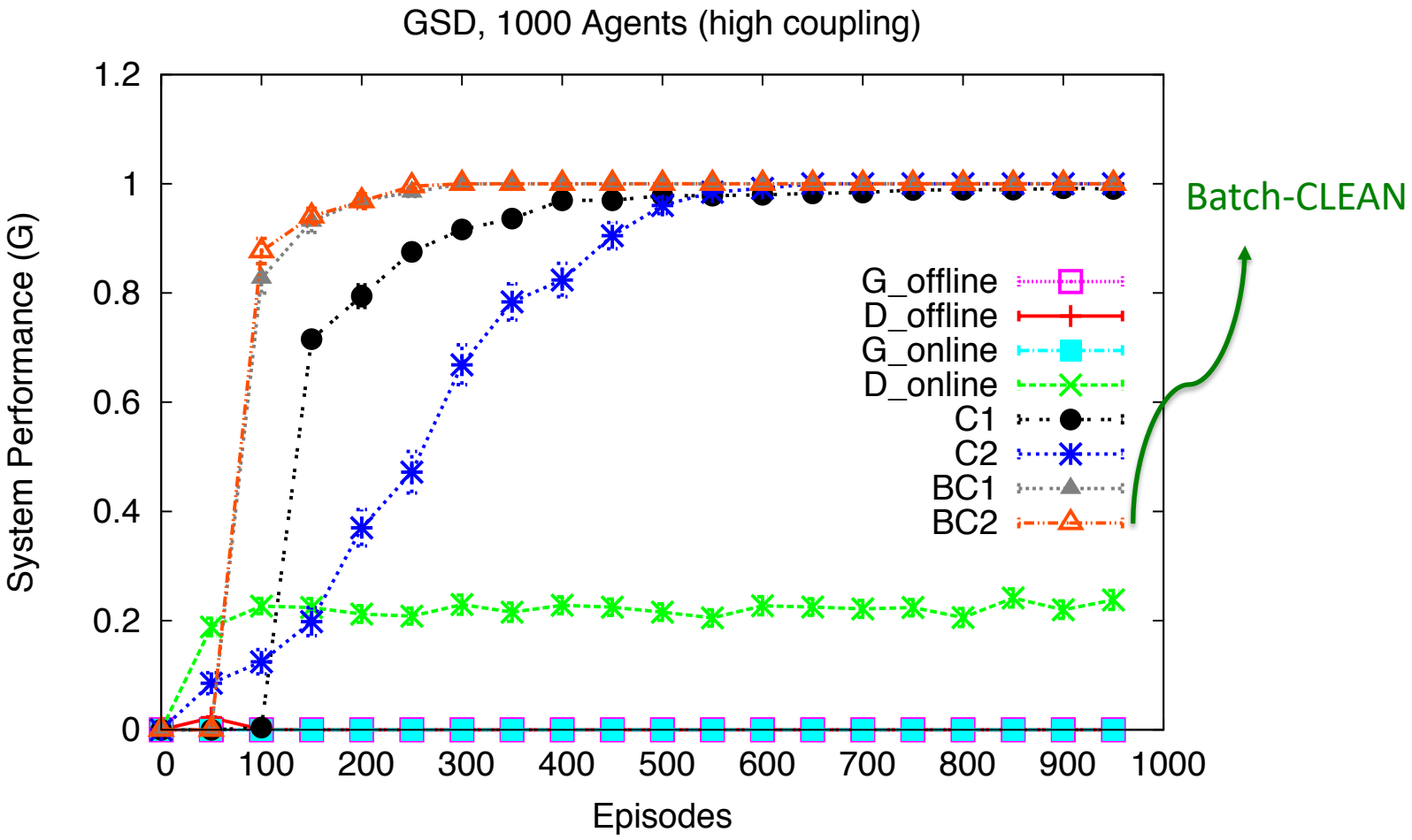
$$G(x) = xe^{\frac{-(x-\mu)^2}{\sigma^2}} \qquad x = \sum_{i=0}^{n} a_i$$

μ and σ are parameters

  - The goal of the agents is to choose their individual actions a_i in such a way that the sum of their individual actions optimize the objective

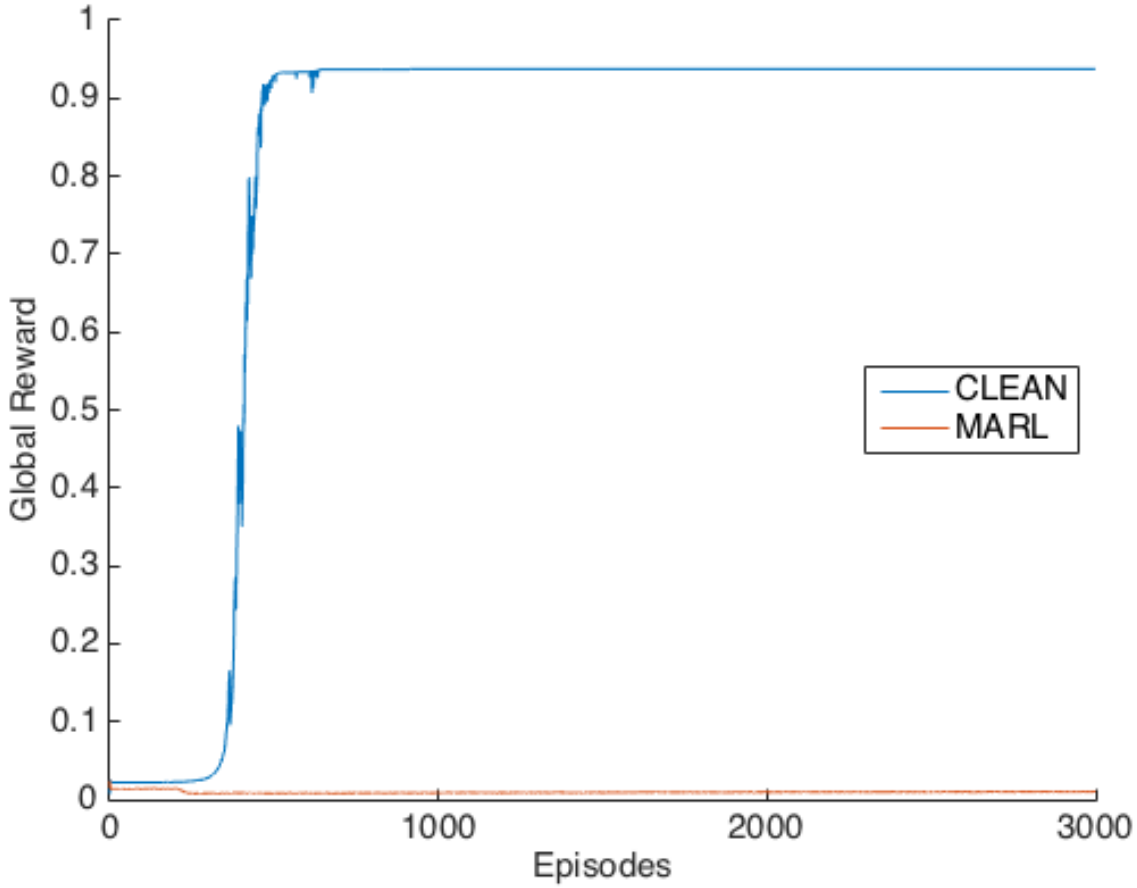
C. Holmesparker, M. E. Taylor, A. K. Agogino, and K. Tumer. Clean rewards to improve coordination by removing exploratory action noise. In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 03, pages 127–134, 2014.

CLEAN rewards: Experiment



C. Holmesparker, M. E. Taylor, A. K. Agogino, and K. Tumer. Clean rewards to improve coordination by removing exploratory action noise. In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 03, pages 127–134, 2014.

CLEAN rewards: Feature selection Experiment



Malialis, Kleanthis, et al. "Feature Selection as a Multiagent Coordination Problem." arXiv preprint arXiv:1603.05152(2016).

Counterfactual Multi-Agent Policy Gradients

- COMA learns a centralised critic Q(s, u) for the joint action u
  - For each agent a, one can compute an advantage function that compares the Q-value for the current action u^a to a counterfactual baseline that marginalises out u^a, while keeping the other agents' actions u^{-a} fixed:

$$A^a(s, \mathbf{u}) = Q(s, \mathbf{u}) - \sum_{u'^a} \pi^a(u'^a|\tau^a) Q(s, (\mathbf{u}^{-a}, u'^a))$$

Foerster, Jakob N., et al. "Counterfactual multi-agent policy gradients." *Thirty-second AAAI conference on artificial intelligence*. 2018.

Counterfactual Multi-Agent Policy Gradients

- In multiagent case, the baseline is unbiased as it marginalizes the target agent's action out
  - The expected the expected contribution of the baseline

$$
\begin{aligned}
g_b = & -\sum_s d^{\boldsymbol{\pi}}(s) \sum_a \sum_{\mathbf{u}^{-a}} \boldsymbol{\pi}(\mathbf{u}^{-a}|\boldsymbol{\tau}-a) \cdot \\
& \sum_{u^a} \pi^a(u^a|\tau^a) \nabla_\theta \log \pi^a(u^a|\tau^a) b(s, \mathbf{u}^{-a}) \\
= & -\sum_s d^{\boldsymbol{\pi}}(s) \sum_a \sum_{\mathbf{u}^{-a}} \boldsymbol{\pi}(\mathbf{u}^{-a}|\boldsymbol{\tau}-a) \cdot \\
& \sum_{u^a} \nabla_\theta \pi^a(u^a|\tau^a) b(s, \mathbf{u}^{-a}) \\
= & -\sum_s d^{\boldsymbol{\pi}}(s) \sum_a \sum_{\mathbf{u}^{-a}} \boldsymbol{\pi}(\mathbf{u}^{-a}|\boldsymbol{\tau}-a) b(s, \mathbf{u}^{-a}) \nabla_\theta 1 \\
= & \, 0.
\end{aligned}
$$

Foerster, Jakob N., et al. "Counterfactual multi-agent policy gradients." *Thirty-second AAAI conference on artificial intelligence*. 2018.

Coordination Games

- Games with more than one pure Nash equilibrium are sometimes called coordination games
  - as if pre-game negotiations are allowed, the players have to agree on one of them

	soccer	ballet
soccer	3, 2	1, 1
ballet	0, 0	2, 3

Battle of the sexes

	Left	Right
Left	10, 10	0, 0
Right	0, 0	10, 10

Pure coordination game

	Stag	Hare
Stag	10, 10	0, 8
Hare	8, 0	7, 7

Stag hunt

- Equilibrium selection: particular equilibria are focal for one reason or another
  - may give higher payoffs, be naturally more salient, may be more fair, or may be safer

Bi-level Coordination Games

- Although the original coordination game model is symmetric that
  - agents should make decision simultaneously
- We can consider the coordination problem from an asymmetric angle
  - There is a leader and a follower (or multiple follower)
  - The following agent observes the actions of the leading agent and always plays the best response
- The Stackelberg equilibrium (SE) is set up as the learning objective rather than the Nash equilibrium (NE)
  - The SE optimizes the leader's policy given that the follower always plays the best- response policy

Zhang, Haifeng, et al. "Bi-level Actor-Critic for Multi-agent Coordination." *AAAI2020*

Bi-level Coordination Games

- The Stackelberg equilibrium (SE) is set up as the learning objective rather than the Nash equilibrium (NE)
  - The SE is Pareto superior than the NE in a wide range of environments

	soccer	ballet
soccer	3, 2	1, 1
ballet	0, 0	2, 3

Battle of the sexes

	Left	Right
Left	10, 10	0, 0
Right	0, 0	10, 10

Pure coordination game

	Stag	Hare
Stag	10, 10	0, 8
Hare	8, 0	7, 7

Stag hunt

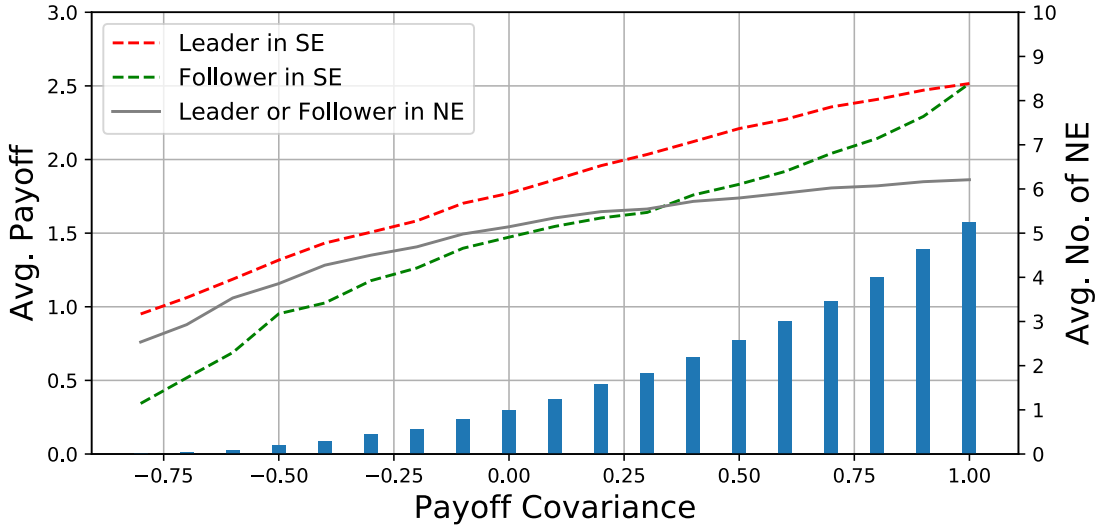
- The SE has the certainty or uniqueness
  - Multiple NE may exist in a game while multiple SE only exist under very strict conditions
    - Multiple SE only exist when given the policy of the leader, multiple policies of the follower achieve the maximal payoff, or given the best response of the follower, multiple policies of the leader achieve the maximal payoff.

Zhang, Haifeng, et al. "Bi-level Actor-Critic for Multi-agent Coordination." *AAAI2020*

Bi-level Coordination Games

- The SE in general is Pareto superior to the average NE in coordination problems, especially in highly cooperative
- To see this,
  - let us define the co-operation level of two-player Markov games as the correlation between the cumulative rewards of the agents:

$$CL = \frac{\sum_{\vec{\pi}}(V_1^{\vec{\pi}} - \bar{V}_1)(V_2^{\vec{\pi}} - \bar{V}_2)}{\sqrt{\sum_{\vec{\pi}}(V_1^{\vec{\pi}} - \bar{V}_1)^2 \sum_{\vec{\pi}}(V_2^{\vec{\pi}} - \bar{V}_2)^2}}$$



Both the leader and the follower achieve higher payoff in the SE not only in fully cooperative games but also in the games with high cooperation level.

Zhang, Haifeng, et al. "Bi-level Actor-Critic for Multi-agent Coordination." *AAAI2020*

Bi-level Q Learning

- Assuming Agent 1 as the leader and Agent 2 as the follower, our problem is formulated as

$$\max_{\pi_1} \quad \mathbb{E}_{r_1^1, r_1^2 \ldots \sim \pi_1, \pi_2} \sum_{t=1}^{\infty} \gamma^t r_1^t$$

$$\text{s.t.} \quad \pi_1 \in \Pi_1$$

$$\max_{\pi_2} \quad \mathbb{E}_{r_2^1, r_2^2 \ldots \sim \pi_1, \pi_2} \sum_{t=1}^{\infty} \gamma^t r_2^t$$

$$\text{s.t.} \quad \pi_2 \in \Pi_2.$$

- The Bellman equation for the joint action:

$$Q_i^*(s, \vec{a}) = R(s, \vec{a}) + \gamma \sum_{s'} P(s, \vec{a}, s') V_i^*(s').$$

$$V_i^*(s) = Stackelberg_i \left(\mathrm{Q}_i^*(s, \vec{a}), \mathrm{Q}_{-i}^*(s, \vec{a}) \right)$$

  - where *Stackelberg* () denotes the i-th agent's payoff in the Stackerberg Equilibrium of the matrix game

Zhang, Haifeng, et al. "Bi-level Actor-Critic for Multi-agent Coordination." *AAAI2020*

Bi-level Q Learning

- Based on the bi-level Bellman equation, we are able to update the Q-values iteratively
- Formally, we have the update rules for Q_1 and Q_2 tables given a transaction $\langle s, a_1, a_2, s', r_1, r_2 \rangle$ with learning rate α_i

$$a_1' \leftarrow \underset{a_1}{\operatorname{argmax}} Q_1(s', a_1, \underset{a_2}{\operatorname{argmax}} Q_2(s', a_1, a_2)),$$

$$a_2' \leftarrow \underset{a_2}{\operatorname{argmax}} Q_2(s', a_1', a_2),$$

$$\begin{aligned} Q_1(s, a_1, a_2) \leftarrow & (1 - \alpha_1) Q_1(s, a_1, a_2) \\ & + \alpha_1 (r_1 + \gamma Q_1(s', a_1', a_2')), \\ Q_2(s, a_1, a_2) \leftarrow & (1 - \alpha_2) Q_2(s, a_1, a_2) \\ & + \alpha_2 (r_2 + \gamma Q_2(s', a_1', a_2')). \end{aligned}$$

Zhang, Haifeng, et al. "Bi-level Actor-Critic for Multi-agent Coordination." *AAAI2020*

Bi-level Actor-Critic

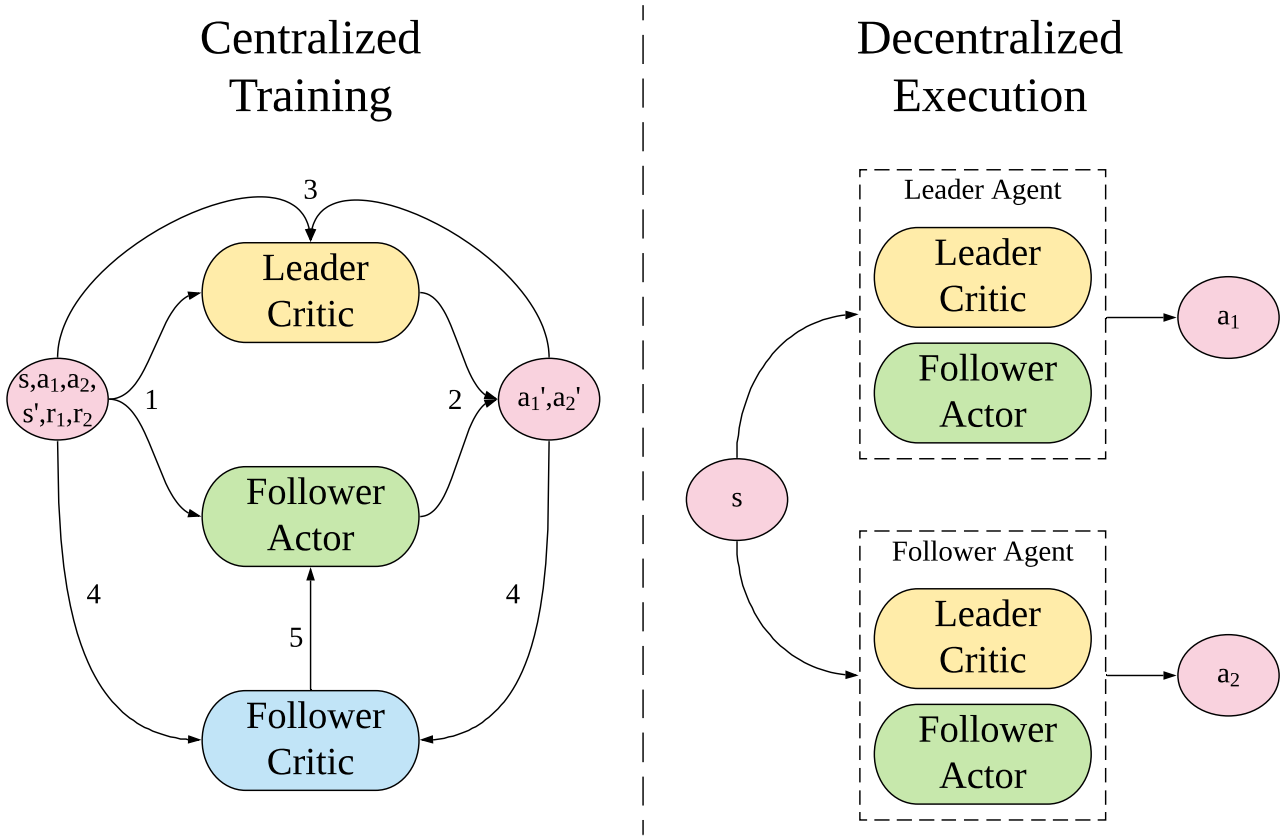
- The bi-level actor-critic (Bi-AC) method introduces an actor for the follower while keeping the leader as a Q-learner
  - Formally, let $\pi_2(s, a_1; \phi 2) \in PD(A_2)$ denote the policy model (or actor) of agent 2, which takes agent 1's action as its input in addition to the current state.
- We have the following update rules given a transaction $\langle s, a_1, a_2, s', r_1, r_2 \rangle$ with learning rate α_i, β:

$$
\begin{aligned}
a_1' &\leftarrow \underset{a_1}{\operatorname{argmax}} Q_1(s', a_1, \pi_2(s', a_1; \phi_2); \theta_1), \\
a_2' &\leftarrow \pi_2(s', a_1'; \phi_2), \\
\delta_i &\leftarrow r_i + \gamma Q_i(s', \vec{a}'; \theta_i) - Q_i(s, \vec{a}; \theta_i), i = 1, 2, \\
\theta_i &\leftarrow \theta_1 + \alpha_i \delta_i \nabla_{\theta_i} Q_i(s, \vec{a}; \theta_i), i = 1, 2, \\
\phi_2 &\leftarrow \phi_2 + \beta \nabla_{\phi_2} \log \pi_2(s, \vec{a}; \phi_2) Q_2(s, \vec{a}; \theta_2).
\end{aligned}
$$

Zhang, Haifeng, et al. "Bi-level Actor-Critic for Multi-agent Coordination." *AAAI2020*

Bi-level Actor-Critic

- Bi-AC uses a centralized-training-decentralized-execution method



Zhang, Haifeng, et al. "Bi-level Actor-Critic for Multi-agent Coordination." *AAAI2020*

Content

- Emergent behaviours
- Agents modeling agents
- Learning communication
- Learning cooperation
- Many-agent learning

Mean field RL

- One of ways to tackling the scalability is to consider *mean-field approximation*, with an extremely large number of homogeneous agents
  - Each agent's effect on the overall multi-agent system can become small
  - All agents being interchangeable/indistinguishable
- The interaction with other agents is captured simply by some mean-field quantity, e.g.,
  - the average state, or the empirical distribution of states
- Each agent only needs to find the best response to the mean-field, which considerably simplifies the analysis.

Zhang, Kaiqing, Zhuoran Yang, and Tamer Başar. "Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms." *arXiv preprint arXiv:1911.10635* (2019).

A Toy Example

- Suppose we have a meeting with a very large number of participants
  - All the data is common knowledge to the meeting participants
  - t the scheduled time of the meeting
  - τ^i the time at which agent i would like to arrive
  - $\tilde{\tau}^i = \tau^i + \sigma^i \tilde{\epsilon}^i$ is the actual arrival time where $\tilde{\epsilon}^i$ is a normal noise with variance 1, specific to agent i. These uncertainties and their intensity differ since some agents come a long way to participate in the meeting and others are very close.
  - We will note m_0 the distribution of σ^i in the population
  - Let T the actual time the meeting will start – we could define it according to the arrival of participants (e.g., 75% of the population)
- Objective: each participant (agent) decides their arrival time τ^i given the following cost to minimize:

$$\tau^i = \operatorname{argmin} \mathbb{E}\left[\alpha[\tilde{\tau}^i - t]_+ + \beta[\tilde{\tau}^i - T]_+ + \gamma[T - \tilde{\tau}^i]_+\right]$$

reputation effect | personal inconvenience | waiting time

α, β, γ>0 are the parameters

Guéant, Olivier, Jean-Michel Lasry, and Pierre-Louis Lions. "Mean field games and applications." *Paris-Princeton lectures on mathematical finance 2010*. Springer, Berlin, Heidelberg, 2011. 205-266.

A Toy Example

- Q: how does each participant (agent) decide their arrival time τ^i given the following cost to minimize:

$$\mathbb{E}\left[\alpha[\tilde{\tau}^i - t]_+ + \beta[\tilde{\tau}^i - T]_+ + \gamma[T - \tilde{\tau}^i]_+\right]$$

reputation effect personal inconvenience waiting time

α, β, γ>0 are the parameters

- The first order condition gives:

$$\alpha\mathcal{N}\left(\frac{\tau^i - t}{\sigma^i}\right) + (\beta + \gamma)\,\mathcal{N}\left(\frac{\tau^i - T}{\sigma^i}\right) = \gamma$$

where *N* is the cumulative distribution function associated to a normal distribution according to the noise $\tilde{\epsilon}^i$. σ^i follows m_0

Since *N* is a strictly monotonic cumulative distribution function and since parameters α, β and γ are positive, the existence and uniqueness of τ^i can be deduced easily.

Guéant, Olivier, Jean-Michel Lasry, and Pierre-Louis Lions. "Mean field games and applications." *Paris-Princeton lectures on mathematical finance 2010*. Springer, Berlin, Heidelberg, 2011. 205-266.

A Toy Example

- We define *F* the cumulative distribution function of the agents' real arrival times $\tilde{\tau}^i$
  - It is due to the uncertainty of σ^i : $\sigma^i \mapsto \tilde{\tau}^i$
- We have the following update steps:

$$T^{**} : T \mapsto (\tau^i(\cdot;T))_i \mapsto (\tilde{\tau}^i(\cdot;T))_i \mapsto F = F(\cdot;T) \mapsto T^*(F)$$

- One can prove that T $**$ is a contraction mapping

Guéant, Olivier, Jean-Michel Lasry, and Pierre-Louis Lions. "Mean field games and applications." *Paris-Princeton lectures on mathematical finance 2010*. Springer, Berlin, Heidelberg, 2011. 205-266.

Mean-field MARL

- *Mean Field Reinforcement Learning*
  - interactions within the population of agents are approximated by those between a single agent and the average effect from neighbouring agents;
  - the interplay between the two entities is mutually reinforced:
    - the learning of the individual agent's optimal policy depends on the dynamics of the population,
    - while the dynamics of the population change according to the collective patterns of the individual policies.

$$Q^j(s, \boldsymbol{a}) \equiv \frac{1}{N^j} \sum_{k \in \mathcal{K}^j} Q^j(s, a^j, a^k),$$

Joint action is replaced by pairwise interactions

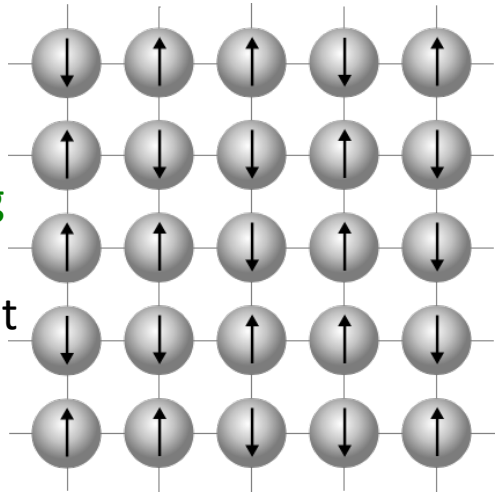
$$Q^j_{t+1}(s, a^j, \bar{a}) = (1 - \alpha_t) Q^j_t(s, a^j, \bar{a}) + \alpha_t \left[r^j_t + \gamma v^j_t(s') \right]$$

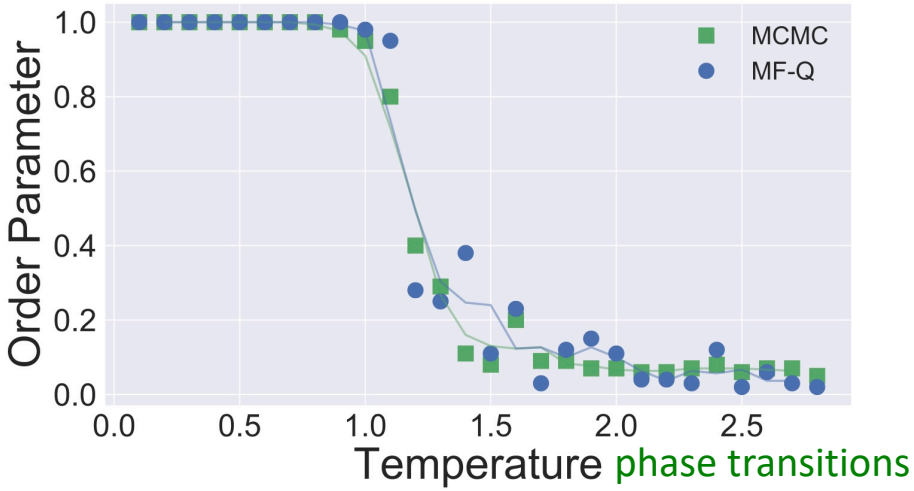
Interplayed with a mean agent

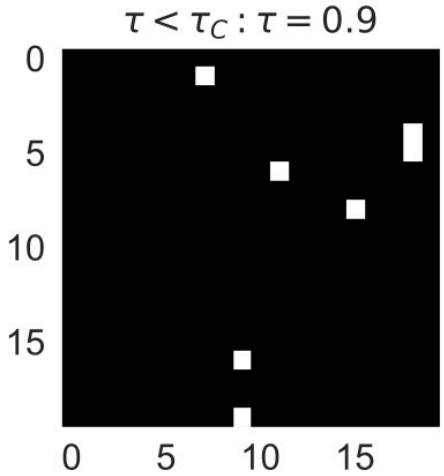
Yang Y, Luo R, Li M, Zhou M, Zhang W, Wang J. Mean Field Multi-Agent Reinforcement Learning. arXiv preprint arXiv:1802.05438. 2018 Feb 15.

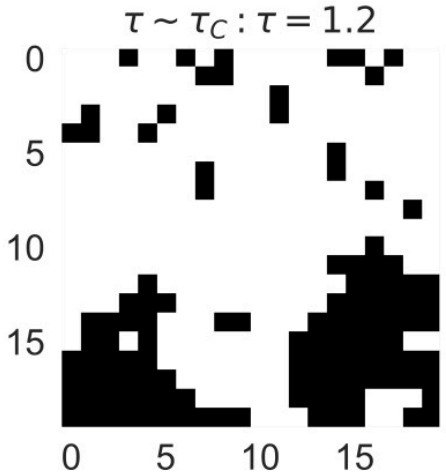
Mean-field MARL: experiments

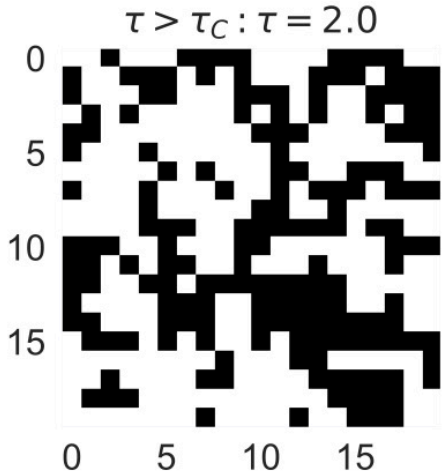
A model-free method to learning the Ising model (atomic spins that can be in one of two states)







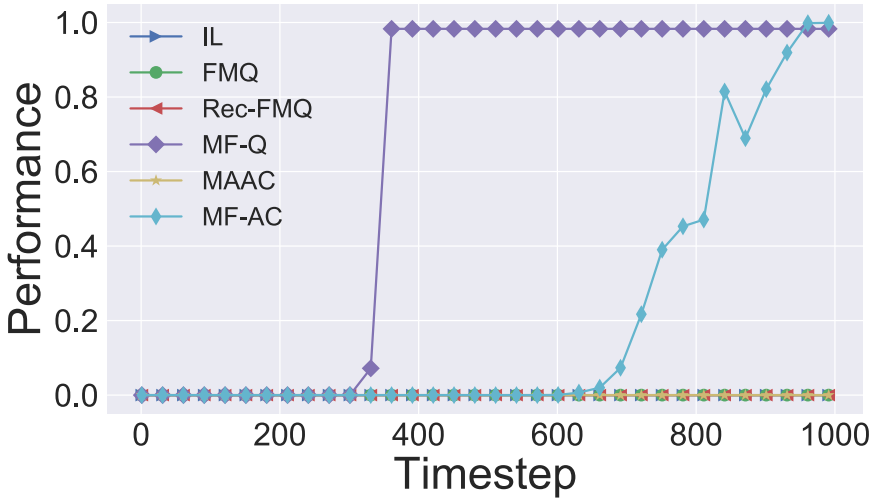


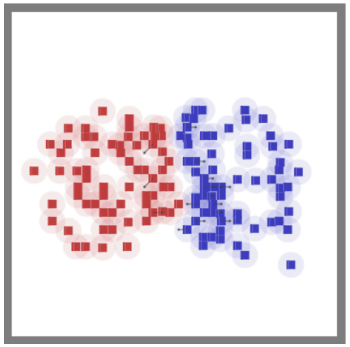


Yang Y, Luo R, Li M, Zhou M, Zhang W, Wang J. Mean Field Multi-Agent Reinforcement Learning. arXiv preprint arXiv:1802.05438. 2018 Feb 15.

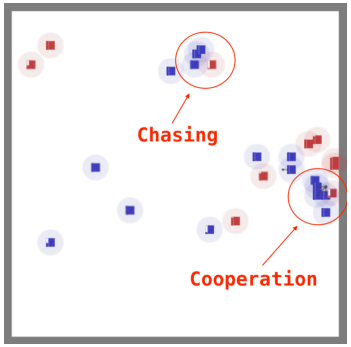
Mean-field MARL: experiments

- The Gaussian Squeeze Domain (GSD):
  - each agent contributes to a system objective
- Battle games:

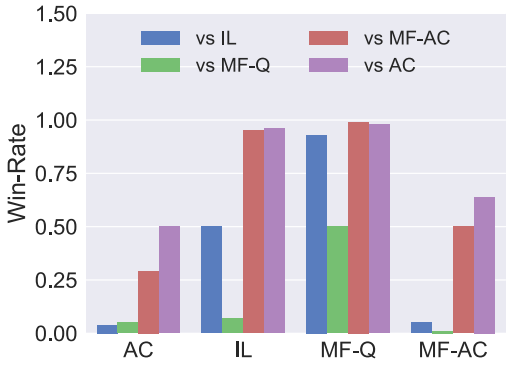




(a) two groups agents



(b) chasing and cooperation



Yang Y, Luo R, Li M, Zhou M, Zhang W, Wang J. Mean Field Multi-Agent Reinforcement Learning. arXiv preprint arXiv:1802.05438. 2018 Feb 15.

References

- Peng Peng, Quan Yuan, Ying Wen, Yaodong Yang, Zhenkun Tang, Haitao Long, Jun Wang, Multiagent Bidirectionally-Coordinated Nets for Learning to Play StarCraft Combat Games, 2017
- Yaodong Yang , Lantao Yu , Yiwei Bai , Jun Wang , Weinan Zhang , Ying Wen , Yong Yu, , Dynamics of Artificial Populations by Million-agent Reinforcement Learning, 2017
- C. Holmesparker, M. E. Taylor, A. K. Agogino, and K. Tumer. Clean rewards to improve coordination by removing exploratory action noise. In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 03, pages 127–134, 2014.
- Malialis, Kleanthis, et al. "Feature Selection as a Multiagent Coordination Problem." arXiv preprint arXiv:1603.05152(2016).
- Sukhbaatar, Sainbayar, and Rob Fergus. "Learning multiagent communication with backpropagation." *Advances in Neural Information Processing Systems*. 2016.
- Foerster J, Assael IA, de Freitas N, Whiteson S. Learning to communicate with deep multi-agent reinforcement learning. InAdvances in Neural Information Processing Systems 2016 (pp. 2137-2145).