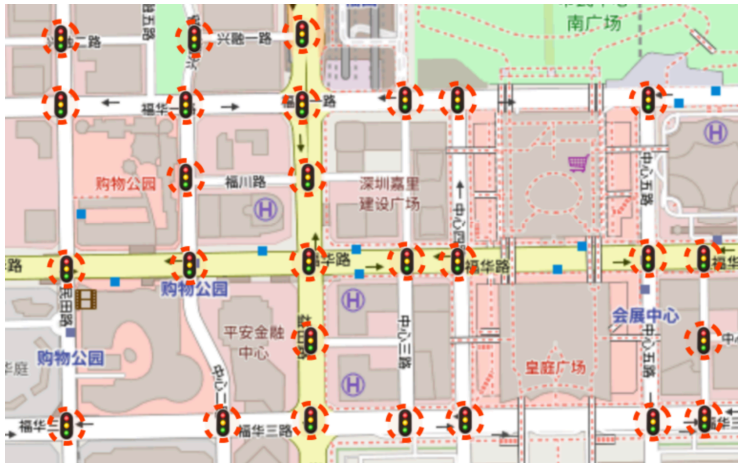
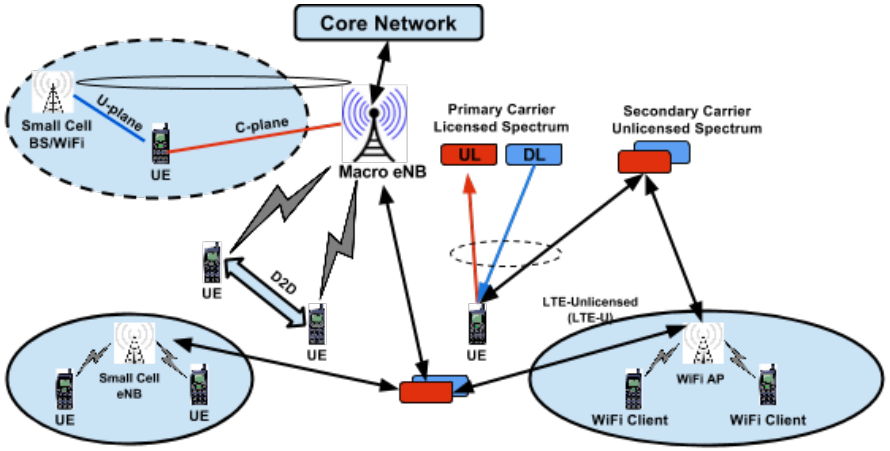
Fully Decentralized Multi-Agent RL

Zongqing Lu
zongqing.lu@pku.edu.cn

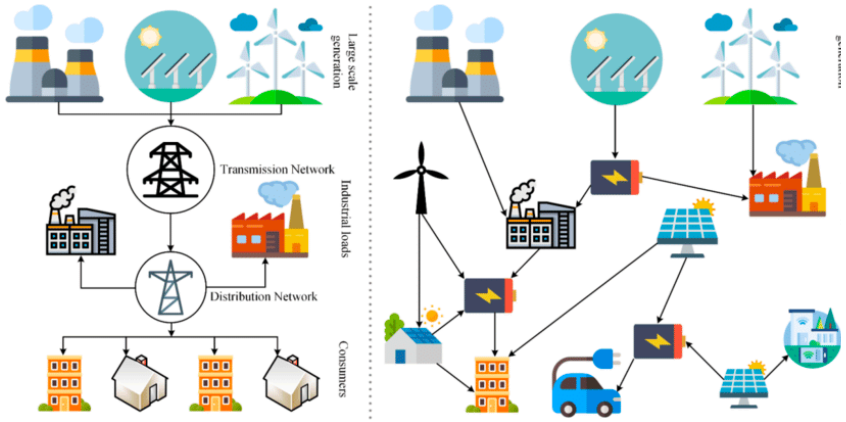
北京大学
PEKING UNIVERSITY



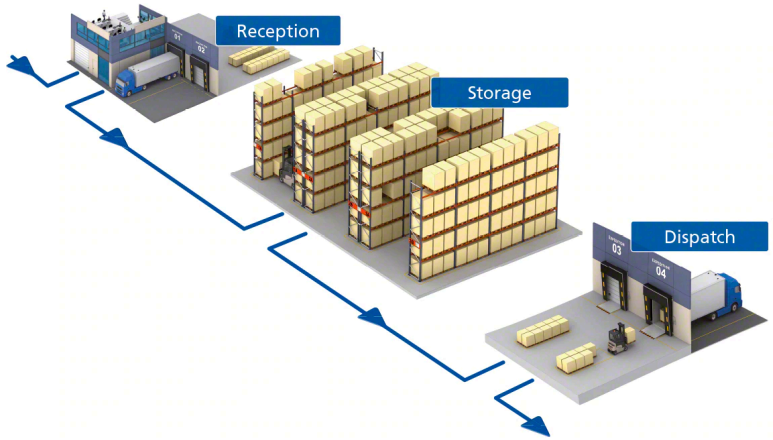
How to control traffic signals in a region to maximize *throughput*?



How to control 5G base stations to optimally *tradeoff coverage and power*?



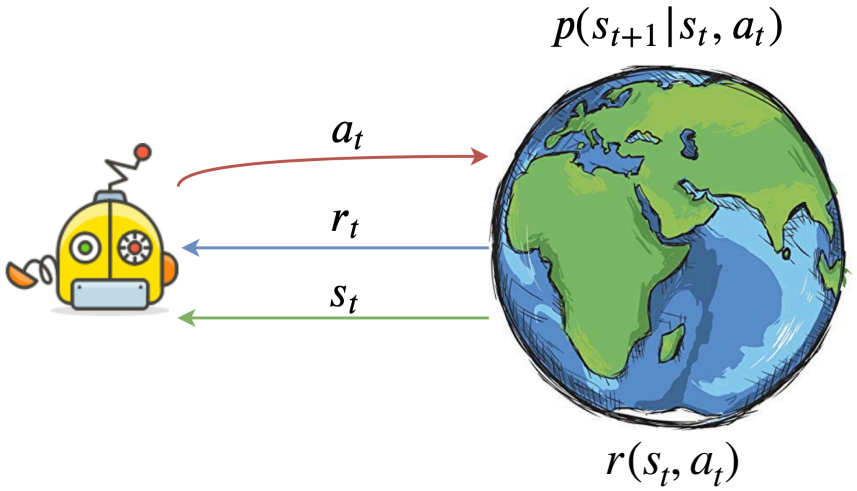
How to control power generators to meet the market demand?



How to optimize supply chain/logistics?

RL and MARL

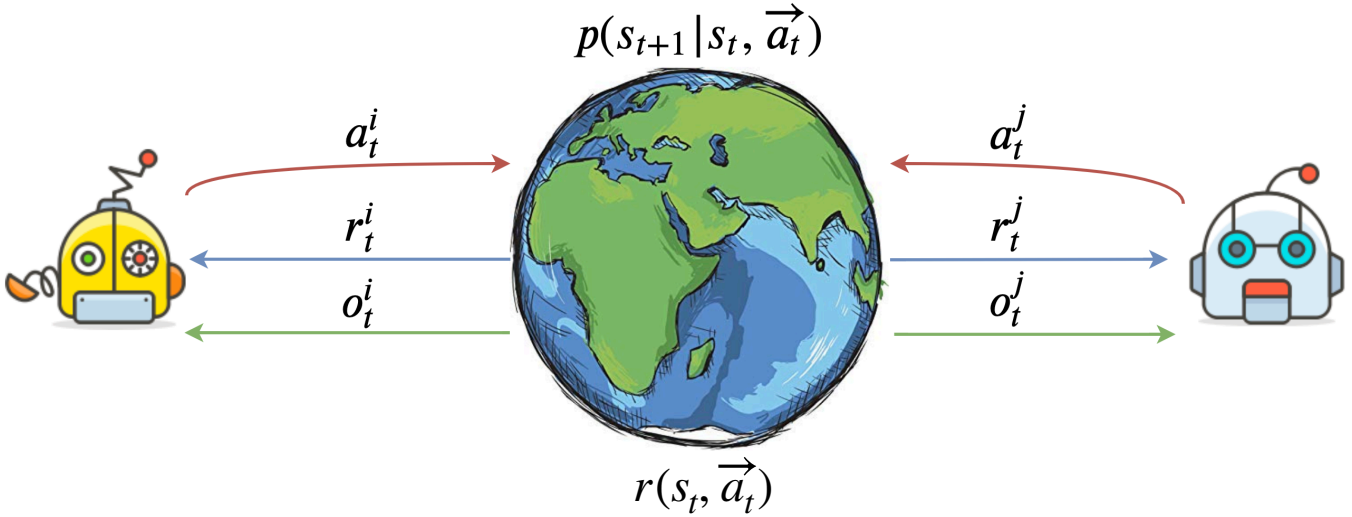
RL



$$\arg\max_{\pi} \mathbb{E}_{\pi} \sum_{t=0}^{\infty} \gamma^t r_t$$

- Q-learning
- Policy gradient (actor-critic)
- Model-based RL

MARL

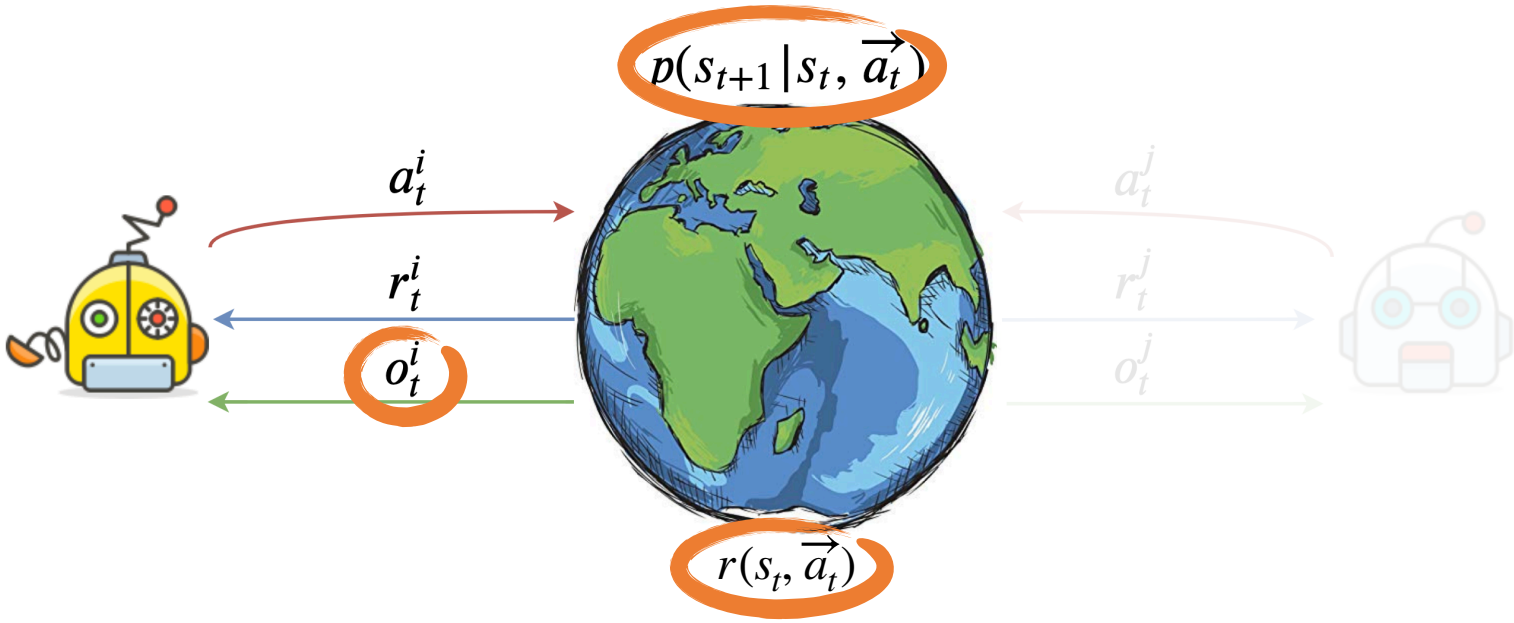


$$r^i = r^j \qquad \arg\max_{\pi} \mathbb{E}_{\pi} \sum_{t=0}^{\infty} \gamma^t r_t$$

Cooperative MARL

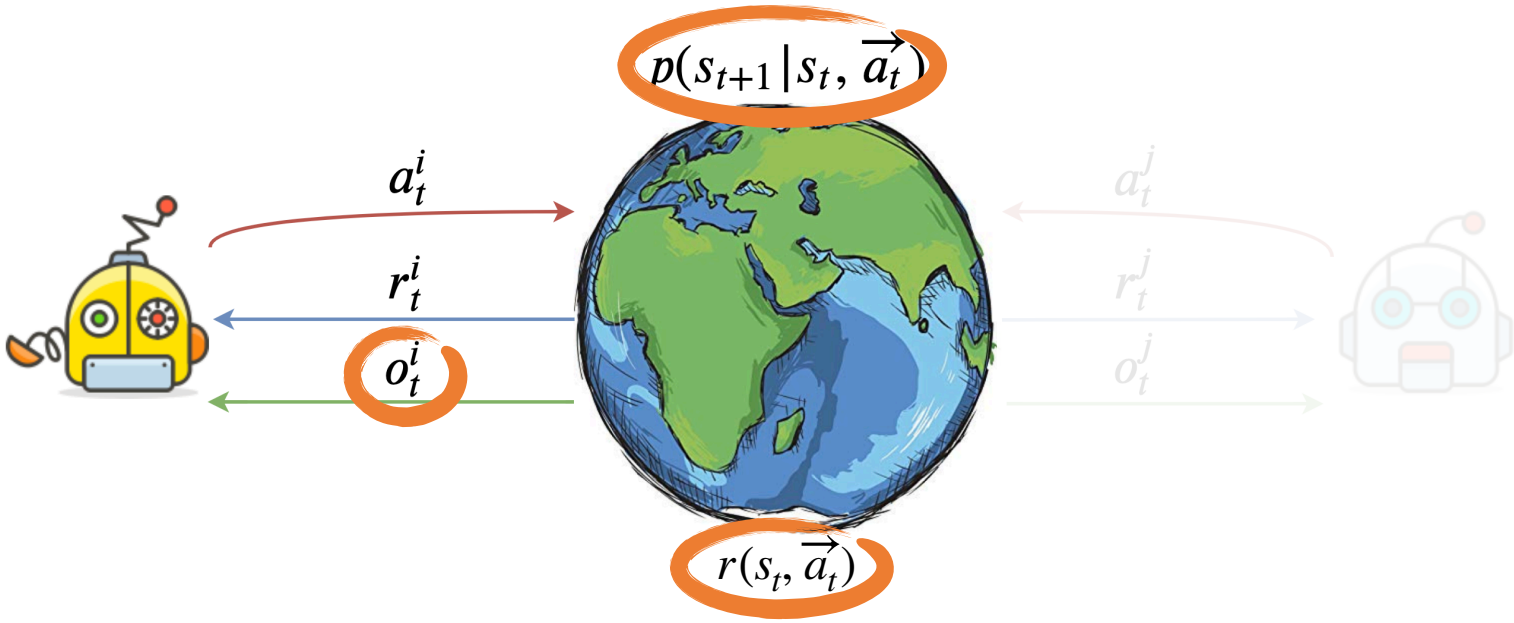
Challenges in cooperative MARL

- Non-stationarity
- Partial observation
- Coordination

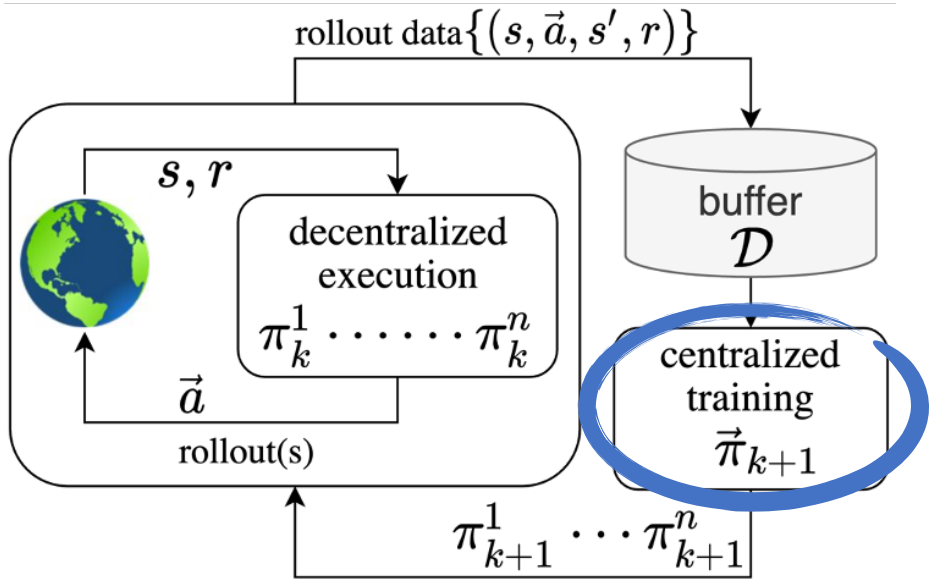


Challenges in cooperative MARL

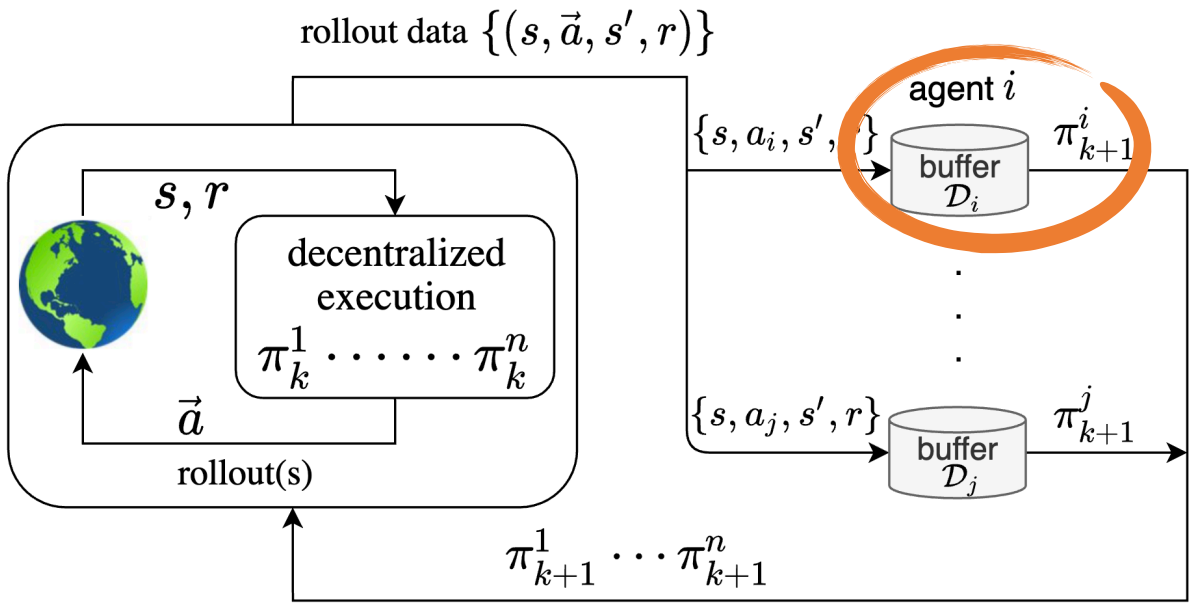
- **Non-stationarity**
- Partial observation
- Coordination



Learning paradigms



Centralized Training with Decentralized Execution (CTDE)



Decentralized Learning

Why should we care decentralized learning?

- Isn’t CTDE enough for MARL?
  - In many cases, there is no centralized entity for training

- **Decentralized learning**
  - Much better scalability and flexibility
  - A old problem, back to the found of MARL
  - ***The way how humans learn!***

Fully decentralized multi-agent RL

- Each agent learns on its **own action**, *without communication or parameter-sharing*

- **Non-stationarity**

$$p(s'|s,\boldsymbol{a}) \rightarrow p(s'|s,a_i) = \sum_{a_{-i}} \pi_{-i}(a_{-i}|s)p(s'|s,a_i,a_{-i})$$

$$r(s,\boldsymbol{a}) \rightarrow r(s,a_i) = \sum_{a_{-i}} \pi_{-i}(a_{-i}|s)r(s,a_i,a_{-i})$$

As π_{-i} *continuously* updates during learning, the environment becomes ***non-stationary***

Simple methods, e.g., independent Q-learning/PPO, have ***no convergence guarantee!***

Outline

1. **Fully decentralized Q-learning**
2. **Fully decentralized policy optimization**

✓ Su et al., A Minimalist Approach to Decentralized Multi-Agent Reinforcement Learning, 2022
✓ Jiang and Lu, I2Q: A fully decentralized Q-learning algorithm, 2022
✓ Su and Lu, Fully decentralized policy optimization, 2022
✓ Luo, Jiang, and Lu, Model-based decentralized policy optimization, 2022

IQL

repeat
  all agents interact in the environment
  for $i \leftarrow 1, n$ **do**
    agent i updates by Q-learning
  end for
until terminate

✘ no convergence guarantee

MA2QL

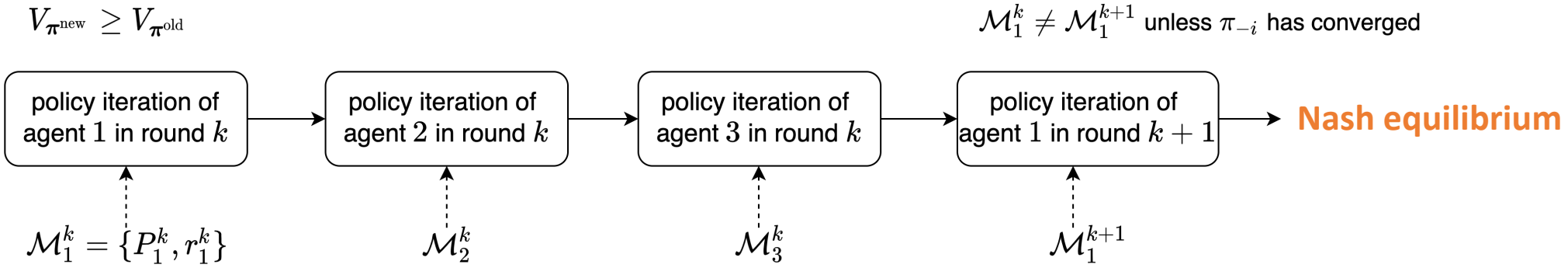
repeat
  for $i \leftarrow 1, n$ **do**
    all agents interact in the environment
    agent i updates by Q-learning
  end for
until terminate

Will this guarantee convergence?

Su et al., A Minimalist Approach to Decentralized Multi-Agent Reinforcement Learning, 2022

A simplest fix to non-stationarity

- Fix the policies of other agents while an agent is learning



Multi-agent alternate policy iteration

✔ convergence guarantee

✘ one-step policy improvement may take very long to converge

$$\pi_i^{*,k} = \arg\max_{\pi_i} \mathbb{E}_{\pi_{-i}^k}\left[Q_{\pi_i,\pi_{-i}^k}(s, a_i, a_{-i})\right]$$

↓

Q-iteration

Su et al., A Minimalist Approach to Decentralized Multi-Agent Reinforcement Learning, 2022

ε-convergent Q-iteration

$$\pi_i^{*,k} = \arg\max_{\pi_i} \mathbb{E}_{\pi_{-i}^k}\left[Q_{\pi_i,\pi_{-i}^k}(s, a_i, a_{-i})\right]$$

Q-iteration

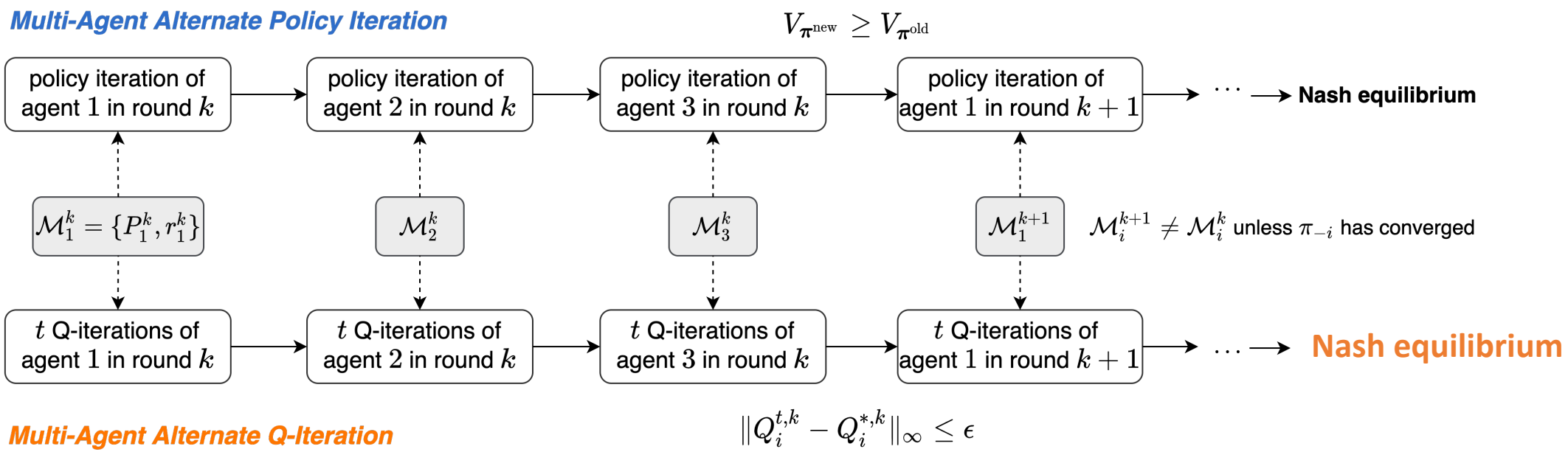
✘ although off-policy at each turn, still asymptotic convergence

As each agent updates the same Q-function at different turns, can we ***truncate*** Q-iteration at each turn?

$$\left\|Q_i^{t,k} - Q_i^{*,k}\right\|_\infty \leq \varepsilon, \quad \text{when} \quad t \geq \frac{\log\left((1-\gamma)\varepsilon\right) - \log(2R + 2\varepsilon)}{\log\gamma}$$

Su et al., A Minimalist Approach to Decentralized Multi-Agent Reinforcement Learning, 2022

Multi-agent alternate Q-iteration



If all agents in turn take Q-iteration to $\|Q_i^k - Q_i^{*,k}\|_\infty \leq \varepsilon$, then their joint policy sequence $\{\pi^k\}$ converges to a Nash equilibrium, where $\pi_i^k(s) = \arg\max_{a_i} Q_i^k(s, a_i)$.

Su et al., A Minimalist Approach to Decentralized Multi-Agent Reinforcement Learning, 2022

MA2QL vs. IQL

IQL

1: **repeat**
2: all agents interact in the environment
3: **for** $i \leftarrow 1, n$ **do**
4: agent i updates by Q-learning
5: **end for**
6: **until** terminate

✘ no convergence guarantee

MA2QL

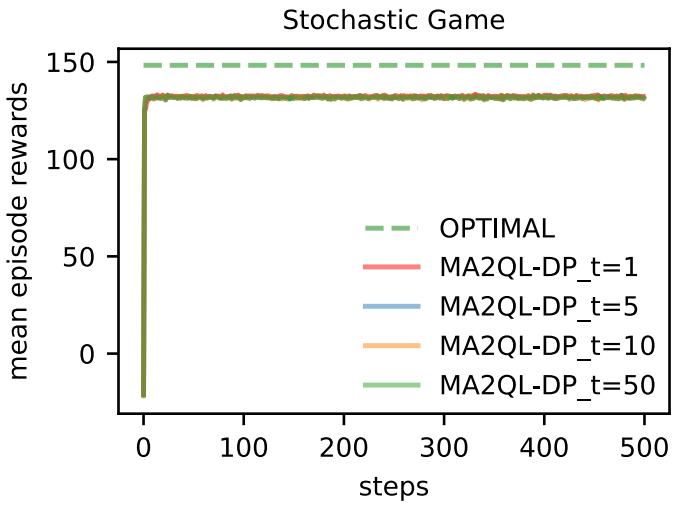
1: **repeat**
2: **for** $i \leftarrow 1, n$ **do**
3: all agents interact in the environment
4: agent i updates by Q-learning
5: **end for**
6: **until** terminate

✔ guaranteed convergence to Nash equilibrium

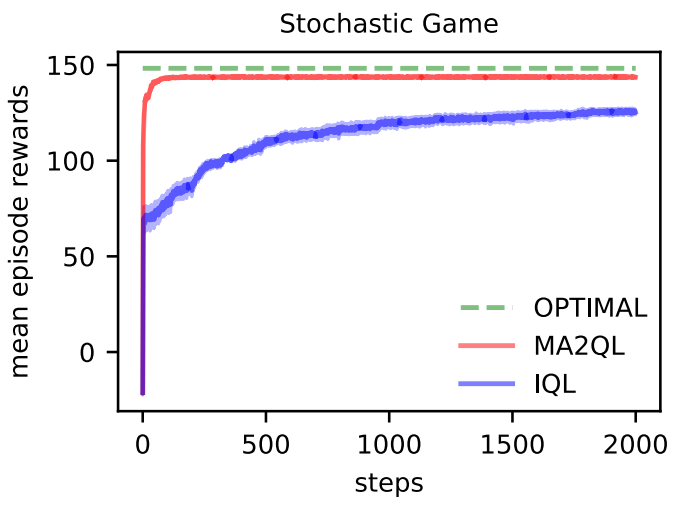
Su et al., A Minimalist Approach to Decentralized Multi-Agent Reinforcement Learning, 2022

MA2QL vs. IQL

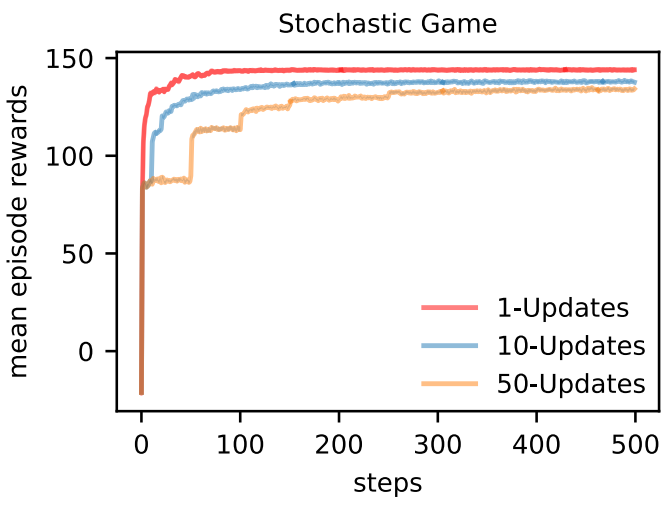
- A didactic game (tabular case)



Q-iteration



MA2QL



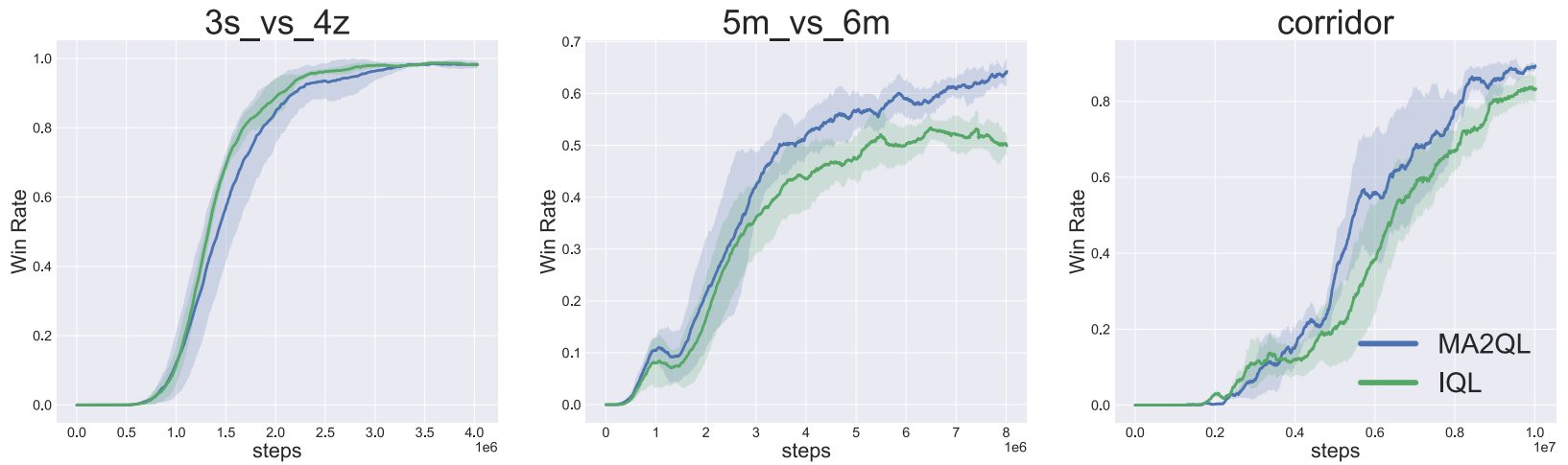
Q-table update

Su et al., A Minimalist Approach to Decentralized Multi-Agent Reinforcement Learning, 2022

MA2QL vs. IQL

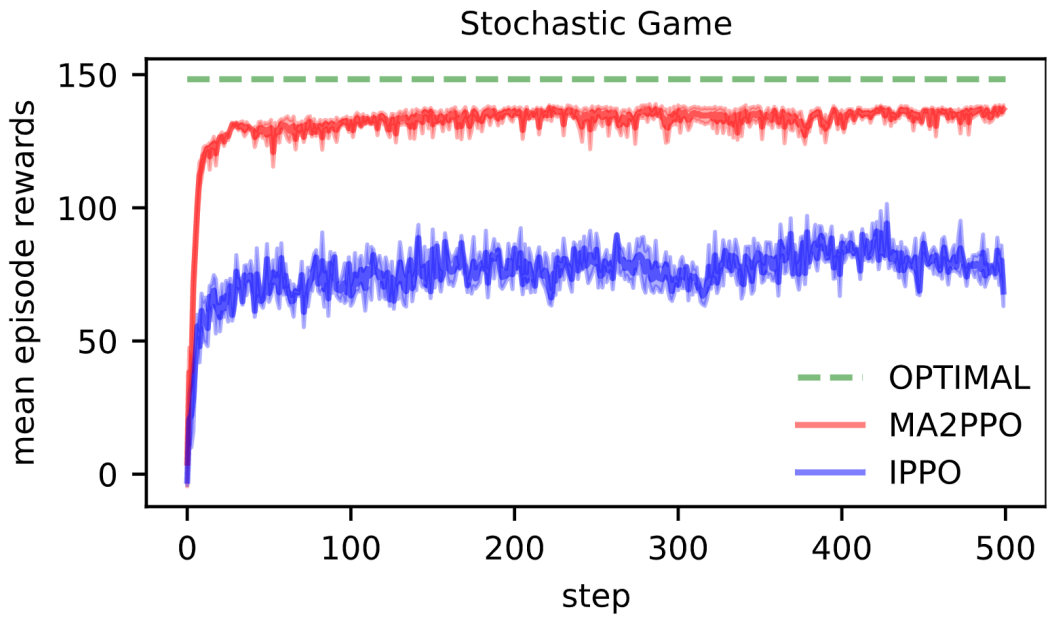
Muti-agent MuJoCo

StarCraft



Su et al., A Minimalist Approach to Decentralized Multi-Agent Reinforcement Learning, 2022

Alternate PPO vs PPO



Su et al., A Minimalist Approach to Decentralized Multi-Agent Reinforcement Learning, 2022

MA2QL

MA2QL

1: **repeat**
2: **for** $i \leftarrow 1, n$ **do**
3: all agents interact in the environment
4: agent i updates by Q-learning
5: **end for**
6: **until** terminate

✔ guaranteed convergence to Nash equilibrium

✘ Nash equilibrium does not guarantee the optimality

Can we have a performance guarantee?

Revisit $P_i(s' \mid s, a_i)$

- Is there an optimal transition function $P_i^\star$?
  - Optimal in terms of **what**?

Suppose there is only one optimal policy

$$\boldsymbol{\pi}^\star(s) = \arg\max_{\boldsymbol{a}} Q^\star(s, \boldsymbol{a}) \rightarrow \boldsymbol{\pi}^\star(s) = \{\pi_1^\star, \ldots, \pi_N^\star\}$$

$$\downarrow$$

$$\boldsymbol{\pi}_{-i}^\star(s, a_i) = \arg\max_{\boldsymbol{a}_{-i}} Q^\star(s, a_i, \boldsymbol{a}_{-i})$$

$$\downarrow$$

$$P_i^\star(s' \mid s, a_i) = P\left(s' \mid s, a_i, \boldsymbol{\pi}_{-i}^\star(s, a_i)\right)$$

$$\downarrow$$

$$\max_{a_i} Q_i^\star(s, a_i) = \max_{a_i} \max_{\boldsymbol{a}_{-i}} Q^\star(s, a_i, \boldsymbol{a}_{-i}) = Q^\star(s, \boldsymbol{\pi}^\star(s))$$

The global optimum under $P_i^\star$

Jiang and Lu, I2Q: a fully decentralized Q-learning algorithm, 2022

Can we construct $P_i^\star$?

- Yes, in deterministic environments!
- QSS learning $Q(s,a) \rightarrow Q(s,s')$

$$Q_i^{\mathrm{ss}}(s,s') = r + \gamma \max_{s'' \in \mathcal{N}(s')} Q_i^{\mathrm{ss}}(s',s'') \rightarrow \max_{s'} Q_i^{\mathrm{ss}}(s,s') = \max_{\boldsymbol{a}} Q^\star(s,\boldsymbol{a})$$

$$s'^\star = \arg\max_{s' \in \mathcal{N}(s,a_i)} Q_i^{\mathrm{ss}}(s,s') \rightarrow \boldsymbol{P_i^\star}$$

In deterministic environments, all agents will converge to ***the optimal policies***, if each agent i performs Q-learning on the transition function $s'^\star = \arg\max_{s' \in \mathcal{N}(s,a_i)} Q_i^{\mathrm{ss}}(s,s')$.

Edwards et al., Estimating Q(s,s') with Deep Deterministic Dynamics Gradients, ICML 2020

Jiang and Lu, I2Q: a fully decentralized Q-learning algorithm, 2022

Ideal independent Q-learning (I2Q)

- Each agent has three modules

① $f(s, a_i)$ approximates $P_i^{\star}\left(s' \mid s, a_i\right)$ by maximizing $\mathbb{E}_{(s,a_i,s')\sim\mathscr{D}_i}\left[\lambda Q_i^{\mathrm{ss}}(s, f_i(s, a_i)) - (f_i(s, a_i) - s')^2\right]$

② $Q(s, s')$ minimizes $\mathbb{E}_{(s,a_i,s',r)\sim\mathscr{D}_i}\left[\left(Q_i^{\mathrm{ss}}(s, s') - r - \gamma \bar{Q}_i^{\mathrm{ss}}(s', f_i(s', a_i'^*))\right)^2\right], \quad a_i'^* = \arg\max_{a_i'} Q_i(s', a_i')$

③ $Q(s, a_i)$ minimizes $\mathbb{E}_{(s,a_i,r)\sim\mathscr{D}_i}\left[\left(Q_i\left(s, a_i\right) - r - \gamma \max_{a_i'} \bar{Q}_i(f_i(s, a_i), a_i')\right)^2\right]$

Jiang and Lu, I2Q: a fully decentralized Q-learning algorithm, 2022

Learning algorithm

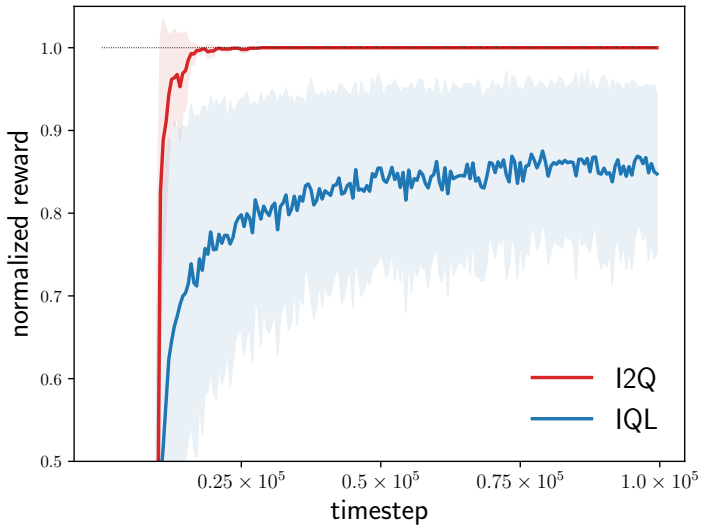
Algorithm 1. I2Q for each agent i

Initialize transition model f_i, Q-networks Q_i and Q_i^{ss}, and the target networks $\bar{Q}_i$ and $\bar{Q}_i^{\text{ss}}$.
Initialize the replay buffer $\mathcal{D}_i$.
for $t = 1, \ldots, max\_iteration$ **do**
    All agents interact in the environment and store experiences (s, a_i, s', r) in replay buffer $\mathcal{D}_i$
    Sample a mini-batch from $\mathcal{D}_i$
    Update f_i by maximizing ①
    Update Q_i^{ss} by minimizing ②
    Update Q_i by minimizing ③
    Update the target networks $\bar{Q}_i$ and $\bar{Q}_i^{\text{ss}}$
end for

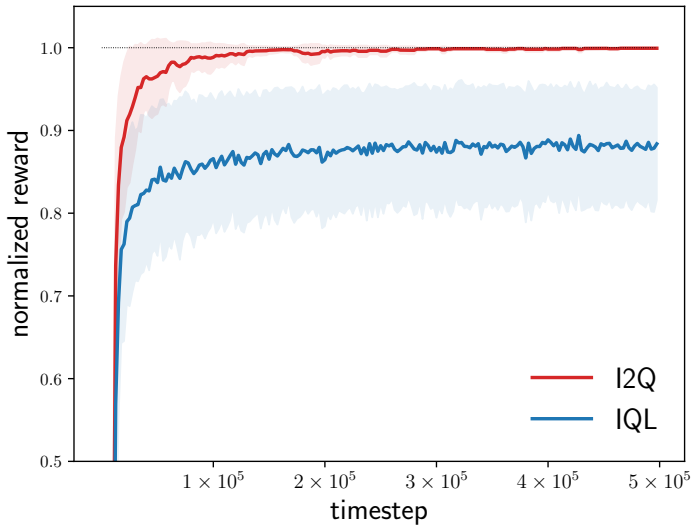
Jiang and Lu, I2Q: a fully decentralized Q-learning algorithm, 2022

I2Q

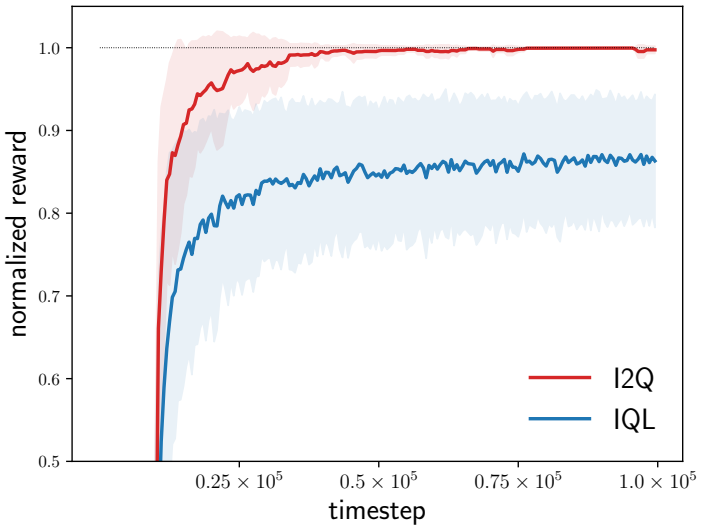
- L-stage matrix game (deterministic tabular case)



$L = 3$



$L = 4$



$L = 5$

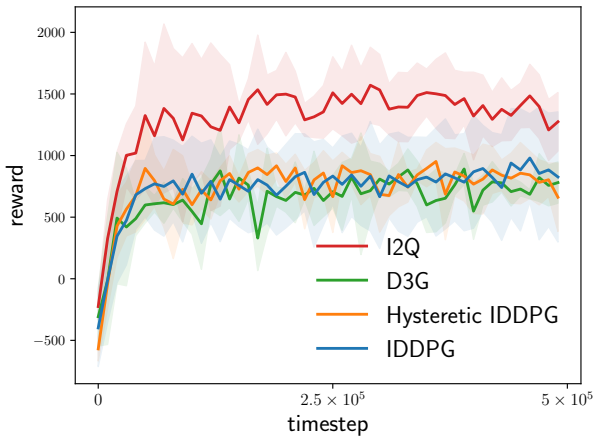
✔ converged to the global optimum

Jiang and Lu, I2Q: a fully decentralized Q-learning algorithm, 2022

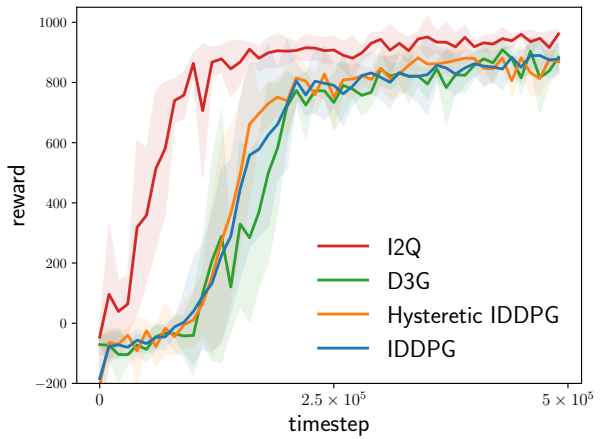
I2Q vs. IQL

Muti-agent MuJoCo
partially observable

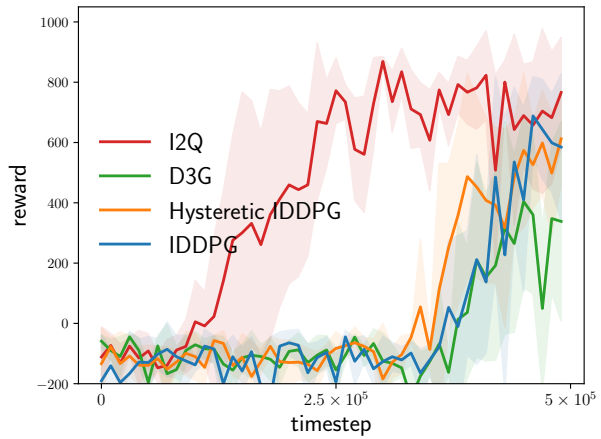
2×3 HalfCheetah



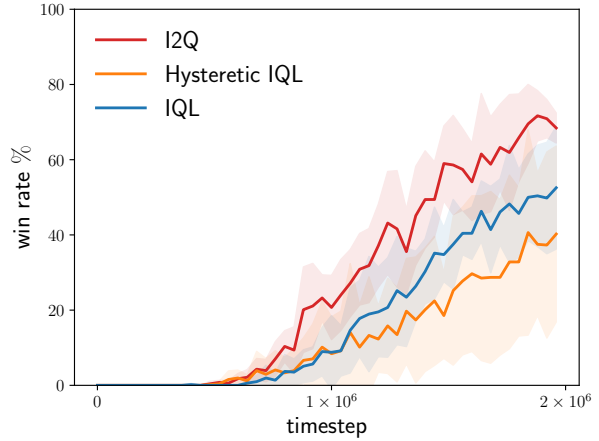
2×4 Ant



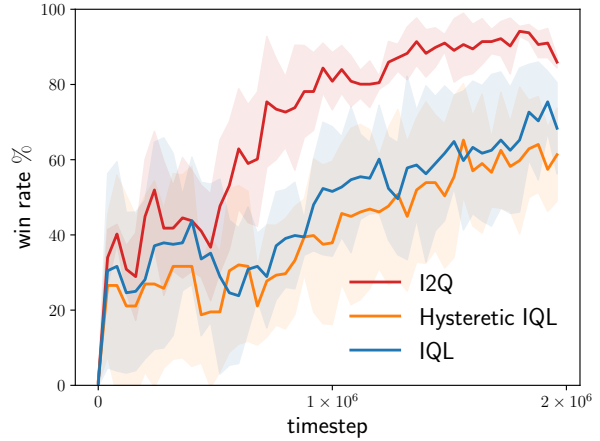
$6 \mid 2$ Ant



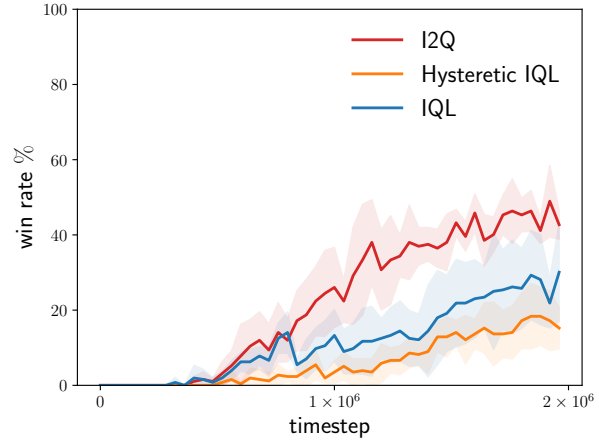
StarCraft
stochastic environment



3s_vs_4z



2s3z



5m_vs_6m

Jiang and Lu, I2Q: a fully decentralized Q-learning algorithm, 2022

Fully decentralized Q-learning

- **Remarks**

✔ MA2QL guarantees Nash equilibrium for learning Q-tables

✔ I2Q guarantees the global optimum in deterministic environments for learning Q-tables

✘ Both are not for learning Q-networks

Can we have a performance guarantee even using neural networks?

Outline

1. **Fully decentralized Q-learning**
2. **Fully decentralized policy optimization**

- ✓ Su et al., A Minimalist Approach to Decentralized Multi-Agent Reinforcement Learning, 2022
- ✓ Jiang and Lu, I2Q: A fully decentralized Q-learning algorithm, 2022
- ✓ Su and Lu, Fully decentralized policy optimization, 2022
- ✓ Luo, Jiang, and Lu, Model-based decentralized policy optimization, 2022

Policy optimization?

- Independent PPO (IPPO) shows strong performance in several benchmarks
  - de Witt et al., 2020
  - Papoudakis et al., 2021
- **Does IPPO have a convergence guarantee?**

Does IPPO have a convergence guarantee?

- Fully decentralized critic

$$Q^{i}_{\pi^{-i}}(s, a_i) = r_{\pi^{-i}}(s, a_i) + \gamma \mathbb{E}_{a_{-i} \sim \pi^{-i}, s' \sim P(\cdot|s,a_i,a_{-i}), a'_i \sim \pi^{i}}[Q^{\pi^{i}}_{\pi^{-i}}(s', a'_i)]$$

$$Q^{\pi^{i}}_{\pi^{-i}}(s, a_i) = \mathbb{E}_{\pi^{-i}}[Q^{\boldsymbol{\pi}}(s, a_i, a_{-i})]$$

$$V^{\pi^{i}}_{\pi^{-i}}(s) = \mathbb{E}_{\pi^{-i}}[V^{\boldsymbol{\pi}}(s)] = V^{\boldsymbol{\pi}}(s)$$

Su and Lu, Fully decentralized policy optimization, 2022

What does IPPO optimize?

- Joint TRPO objective

$$J(\boldsymbol{\pi}_{\text{new}}) - J(\boldsymbol{\pi}_{\text{old}}) \geq \mathscr{L}^{\text{joint}}_{\boldsymbol{\pi}_{\text{old}}}(\boldsymbol{\pi}_{\text{new}}) - C \cdot D^{\max}_{\text{KL}}(\boldsymbol{\pi}_{\text{old}} \| \boldsymbol{\pi}_{\text{new}})$$

where $$\mathscr{L}^{\text{joint}}_{\boldsymbol{\pi}_{\text{old}}}(\boldsymbol{\pi}_{\text{new}}) = \sum_{s} \boldsymbol{\rho}_{\text{old}}(s) \sum_{\boldsymbol{a}} \boldsymbol{\pi}_{\text{new}}(\boldsymbol{a} \mid s) A_{\text{old}}(s, \boldsymbol{a})$$

- IPPO objective

$$\mathscr{L}^{i}_{\boldsymbol{\pi}_{\text{old}}}(\pi^{i}_{\text{new}}) = \sum_{s} \boldsymbol{\rho}_{\text{old}}(s) \sum_{a_i} \pi^{i}_{\text{new}}(a_i \mid s) A^{i}_{\text{old}}(s, a_i)$$

where $$A^{i}_{\text{old}}(s, a_i) = = \mathbb{E}_{\pi^{-i}_{\text{old}}}[A_{\text{old}}(s, a_i, a_{-i})]$$

✘ optimizing this objective does not *necessarily* guarantee the improvement of the joint policy

Su and Lu, Fully decentralized policy optimization, 2022

What should we optimize?

- This is the only objective for fully decentralized learning

$$\mathscr{L}^{i}_{\boldsymbol{\pi}_{\text{old}}}(\pi^{i}_{\text{new}}) = \sum_{s} \boldsymbol{\rho}_{\text{old}}(s) \sum_{a_i} \pi^{i}_{\text{new}}(a_i \mid s) A^{i}_{\text{old}}(s, a_i)$$

- A novel lower bound

$$J(\boldsymbol{\pi}_{\text{new}}) - J(\boldsymbol{\pi}_{\text{old}}) \geq \mathscr{L}^{\text{joint}}_{\boldsymbol{\pi}_{\text{old}}}(\boldsymbol{\pi}_{\text{new}}) - C \cdot D^{\max}_{\text{KL}}(\boldsymbol{\pi}_{\text{old}} \| \boldsymbol{\pi}_{\text{new}})$$

$$J(\boldsymbol{\pi}_{\text{new}}) - J(\boldsymbol{\pi}_{\text{old}}) \geq \frac{1}{N} \sum_{i=1}^{N} \mathscr{L}^{i}_{\boldsymbol{\pi}_{\text{old}}}(\pi^{i}_{\text{new}}) - \tilde{M} \cdot \sum_{i=1}^{N} \sqrt{D^{\max}_{\text{KL}}(\pi^{i}_{\text{old}} \| \pi^{i}_{\text{new}})} - C \cdot \sum_{i=1}^{N} D^{\max}_{\text{KL}}(\pi^{i}_{\text{old}} \| \pi^{i}_{\text{new}})$$

Su and Lu, Fully decentralized policy optimization, 2022

Fully decentralized surrogate

$$J(\boldsymbol{\pi}_{\text{new}}) - J(\boldsymbol{\pi}_{\text{old}}) \geq \frac{1}{N}\sum_{i=1}^{N} \mathscr{L}^{i}_{\boldsymbol{\pi}_{\text{old}}}(\pi^{i}_{\text{new}}) - \tilde{M} \cdot \sum_{i=1}^{N} \sqrt{D^{\max}_{\text{KL}}(\pi^{i}_{\text{old}} \| \pi^{i}_{\text{new}})} - C \cdot \sum_{i=1}^{N} D^{\max}_{\text{KL}}(\pi^{i}_{\text{old}} \| \pi^{i}_{\text{new}})$$

$$\pi^{i}_{\text{new}} = \arg\max_{\pi^{i}} \left(\frac{1}{N} \mathscr{L}^{i}_{\boldsymbol{\pi}_{\text{old}}}(\pi^{i}) - \tilde{M} \cdot \sqrt{D^{\max}_{\text{KL}}(\pi^{i}_{\text{old}} \| \pi^{i})} - C \cdot D^{\max}_{\text{KL}}(\pi^{i}_{\text{old}} \| \pi^{i}) \right)$$

✔ **the joint policy of agents improves monotonically and converges to *suboptimum***

✘ **the step size of the policy update will be small, similar to TRPO**

Su and Lu, Fully decentralized policy optimization, 2022

Fully decentralized policy optimization (DPO)

- The practical algorithm

$$\pi^i_{\text{new}} = \arg\max_{\pi^i} \left(\frac{1}{N} \mathscr{L}^i_{\boldsymbol{\pi}_{\text{old}}}(\pi^i) - \tilde{M} \cdot \sqrt{D^{\max}_{\text{KL}}(\pi^i_{\text{old}} \| \pi^i)} - C \cdot D^{\max}_{\text{KL}}(\pi^i_{\text{old}} \| \pi^i) \right)$$

In practice

$$\pi^i_{\text{new}} = \arg\max_{\pi^i} \left(\frac{1}{N} \mathscr{L}^i_{\boldsymbol{\pi}_{\text{old}}}(\pi^i) - \beta^i_1 \sqrt{D^{\text{avg}}_{\text{KL}}(\pi^i_{\text{old}} \| \pi^i)} - \beta^i_2 D^{\text{avg}}_{\text{KL}}(\pi^i_{\text{old}} \| \pi^i) \right)$$

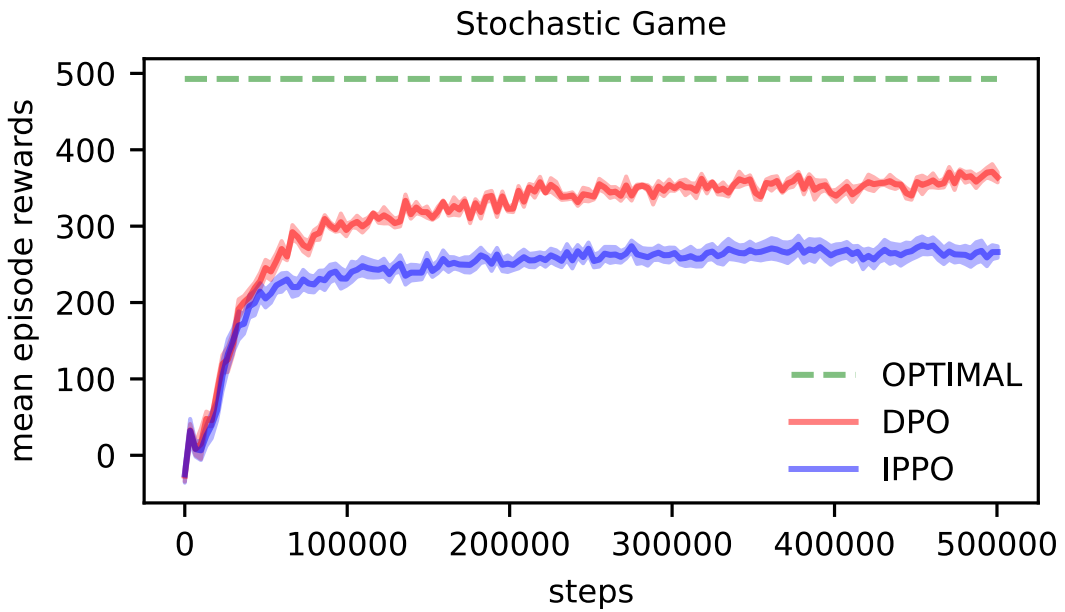
If $D^{\text{avg}}_{\text{KL}}(\pi^i_{\text{old}} \| \pi^i_{\text{new}}) > d_{target} * \delta$, then $\beta^i_j \leftarrow \beta^i_j * \omega \quad \forall j \in \{1,2\}$

If $D^{\text{avg}}_{\text{KL}}(\pi^i_{\text{old}} \| \pi^i_{\text{new}}) < d_{target} / \delta$, then $\beta^i_j \leftarrow \beta^i_j / \omega \quad \forall j \in \{1,2\}$

Su and Lu, Fully decentralized policy optimization, 2022

DPO vs. IPPO

- Tabular case



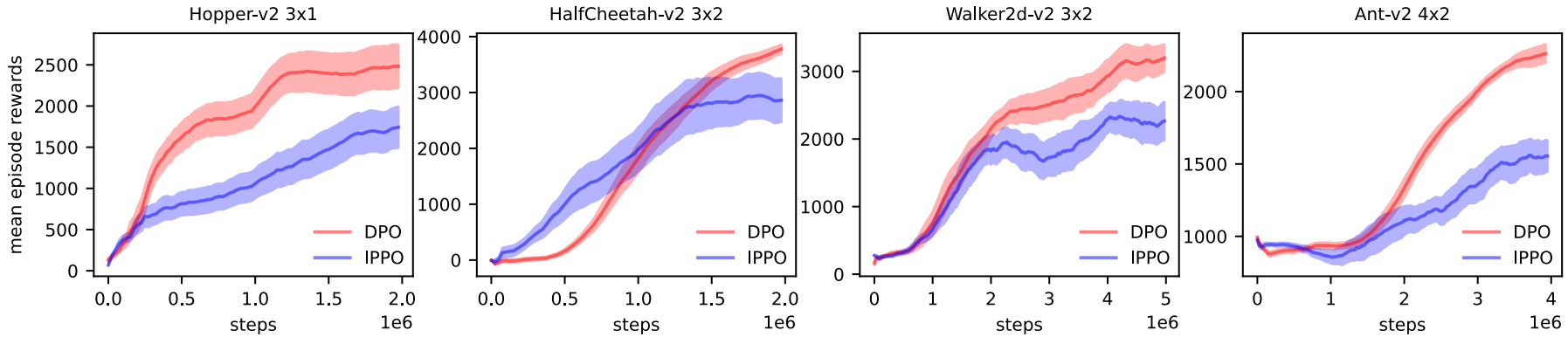
✔ **better sub-optimum empirically**

✘ **still a large gap to the optimum**

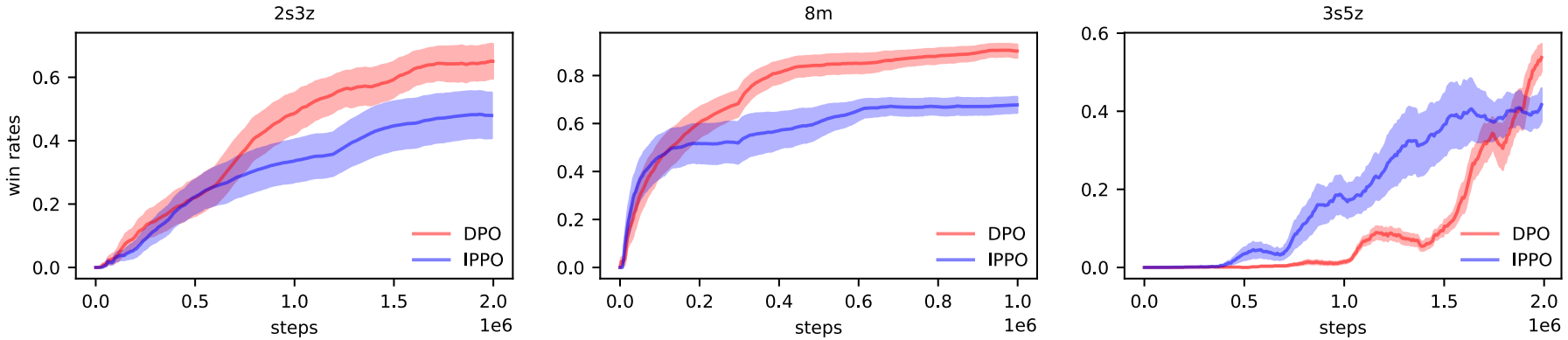
Su and Lu, Fully decentralized policy optimization, 2022

DPO vs. IPPO

Muti-agent MuJoCo



StarCraft



Su and Lu, Fully decentralized policy optimization, 2022

DPO

- **Remarks**

✔ DPO guarantees monotonic improvement and sub-optimum ?

✘ The assumption of stationary discounted state distribution ***may not*** hold in multi-agent settings

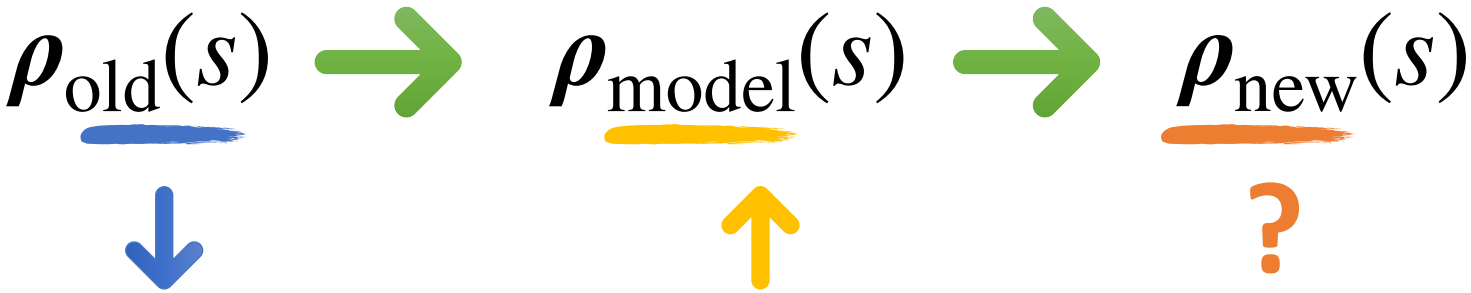
$$\mathscr{L}^{i}_{\boldsymbol{\pi}_{\text{old}}}(\pi^{i}_{\text{new}}) = \sum_{s} \rho_{\text{old}}(s) \sum_{a_i} \pi^{i}_{\text{new}}(a_i \mid s) A^{i}_{\text{old}}(s, a_i)$$

$\rho_{\text{new}}(s)$

Can we bridge the gap between $\rho_{\text{old}}(s)$ and $\rho_{\text{new}}(s)$?

Luo, Jiang, and Lu, Model-based decentralized policy optimization, 2022

Can we bridge the gap between $\rho_{\text{old}}(s)$ and $\rho_{\text{new}}(s)$?



Bridging the gap by learning from experiences generated by the environment model?

Luo, Jiang, and Lu, Model-based decentralized policy optimization, 2022

What model can we have?

- **Fully decentralized model with latent variable**

$$P_i(o_i' \mid o_i, a_i, z_i) \qquad R(o_i, a_i, z_i)$$

z_i captures the inaccessible information of state and other agents' policies

$$\psi_\omega(\cdot \mid o) \rightarrow \psi(\cdot \mid o)$$ **true latent variable function**

Luo, Jiang, and Lu, Model-based decentralized policy optimization, 2022

Does such a simple model work?

$\psi_\omega(\cdot|o)$ → $\psi(\cdot|o)$ true latent variable function

$$\|\Delta\rho^{(\pi^1,\psi^1),(\pi^2,\psi^2)}\|_1 \le C\left(\|\pi^1-\pi^2\|_\infty + \|\psi^1-\psi^2\|_\infty\right)$$

$$\max_n \|\psi^n-\psi^{n-1}\|_\infty > \max_n \|\psi^n_\omega-\psi^{n-1}_\omega\|_\infty$$

Policy optimization

Latent variable function

More stable policy optimization on model

Luo, Jiang, and Lu, Model-based decentralized policy optimization, 2022

Can we guarantee policy improvement?

$$\left| \eta(\pi, \psi^{n+1}) - \eta^{model}(\pi, \psi_\omega^n) \right| \leq C(\epsilon_\theta, \epsilon_\pi, \epsilon_\omega, \epsilon_\psi)$$

model learning → ϵ_θ: **model error**

policy optimization → ϵ_π: **policy divergence**

model learning → ϵ_ω: **learned latent variable function error**

ϵ_ψ: **true latent variable function divergence**

?

Luo, Jiang, and Lu, Model-based decentralized policy optimization, 2022

Can we guarantee policy improvement?

$$\hat{\epsilon}_{\omega} \triangleq \max_{o} D_{\mathrm{TV}}\left(\psi^{n+1} \| \psi_{\omega}^{n}\right)$$

$$\left|\eta\left(\pi, \psi^{n+1}\right) - \eta^{model}\left(\pi, \psi_{\omega}^{n}\right)\right| \leq C(\epsilon_{\theta}, \epsilon_{\pi}, \hat{\epsilon}_{\omega})$$

Constraining $\hat{\epsilon}_{\omega}$ controls the return bound

Luo, Jiang, and Lu, Model-based decentralized policy optimization, 2022

How to constrain $\hat{\epsilon}_\omega$?

- **Fully decentralized model with latent variable prediction**

$$\hat{\epsilon}_\omega \triangleq \max_o D_{\mathrm{TV}}\left(\psi^{n+1} \| \psi_\omega^n\right)$$

$$\max_o D_{\mathrm{TV}}\left(\psi^l(o) \| f\left(\psi^1(o), \cdots, \psi^{l-1}(o)\right)\right)$$

Luo, Jiang, and Lu, Model-based decentralized policy optimization, 2022

Model-based decentralized policy optimization

①Model learning → $\max_{\theta,\omega_l,\phi} \sum_{j=1}^{l} \left(\mathbb{E}_{(o,a,o',r)\sim\mathcal{D}^j, z\sim\psi_{\omega_j}(o)} \left[P_\theta\left(o' \mid o,a,z\right) - \left(R_\phi\left(o,a,z\right) - r\right)^2 \right] \right)$

②Latent variable prediction → $\max_{\zeta} \mathbb{E}_{o\sim\mathcal{D}^l, z^1\sim\psi_{\omega_1}(o),\cdots,z^l\sim\psi_{\omega_l}(o)} \left[f_\zeta\left(z^l \mid z^1,\cdots,z^{l-1}\right) \right]$

③Model rollout → $z \sim f_\zeta\left(\cdot \mid z^2 \sim \psi_{\omega_2}(o),\cdots,z^l \sim \psi_{\omega_l}(o) \right),\ \hat{o}' \sim P_\theta(o' \mid o_a, z),\ \hat{r} = R_\phi(o,a,z)$

④Policy optimization → **DPO/IPPO on branched model rollout**

Luo, Jiang, and Lu, Model-based decentralized policy optimization, 2022

Learning algorithm

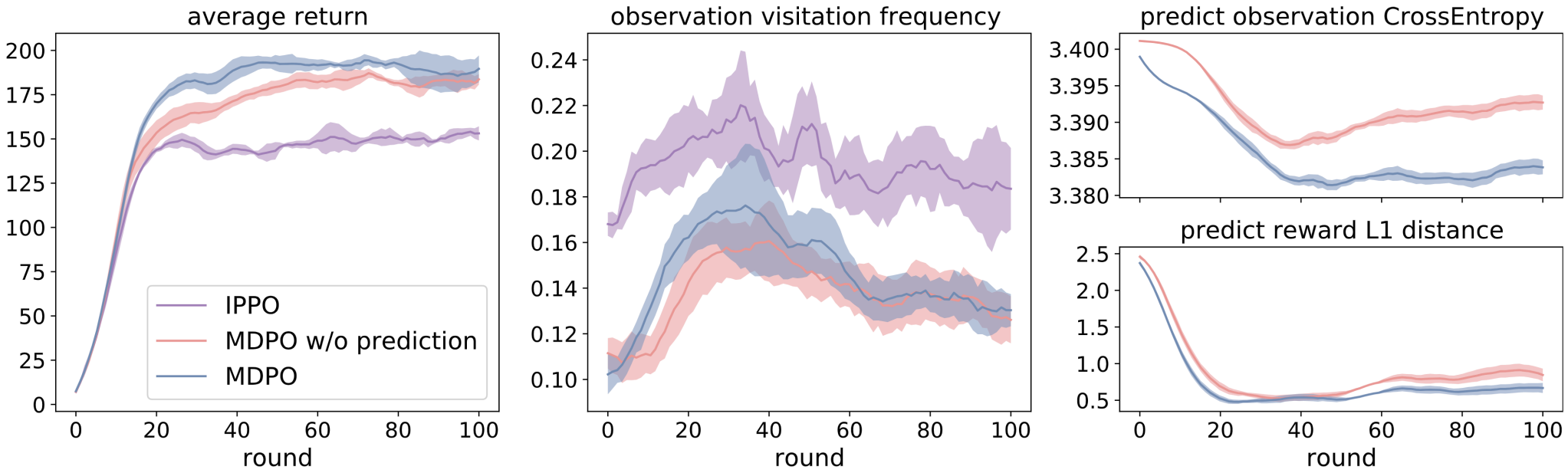
Algorithm 1. MDPO

Initiate $\mathcal{D}_{env} = \{\mathcal{D}^1, \cdots, \mathcal{D}^l\}$, π, P_θ, R_ϕ, $\Psi = \{\psi_{\omega_1}, \cdots, \psi_{\omega_l}\}$, f_ζ.
repeat
    policy rollout in environment and obtain $\mathcal{D}^l$
    optimize P_θ,R_ϕ and ψ_{ω_l} on $\mathcal{D}_{env}$ with ①
    optimize prediction function f_ζ on $\mathcal{D}^l$ with ②
    obtain branched model rollout $\mathcal{D}_{rollout}$ based on $\mathcal{D}^l$ using P_θ, R_ϕ, π, Ψ, and f_ζ with ③
    optimize policy π using $\mathcal{D}_{rollout}$ by IPPO/DPO ④
    for $j \leftarrow 1, \ldots, l-1$ **do**
        $\mathcal{D}^j \leftarrow \mathcal{D}^{j+1}$, $\psi_{\omega_j} \leftarrow \psi_{\omega_{j+1}}$
    end for
until terminate

Luo, Jiang, and Lu, Model-based decentralized policy optimization, 2022

MDPO, MDPO w/o Prediction, and IPPO

- A cooperative stochastic game



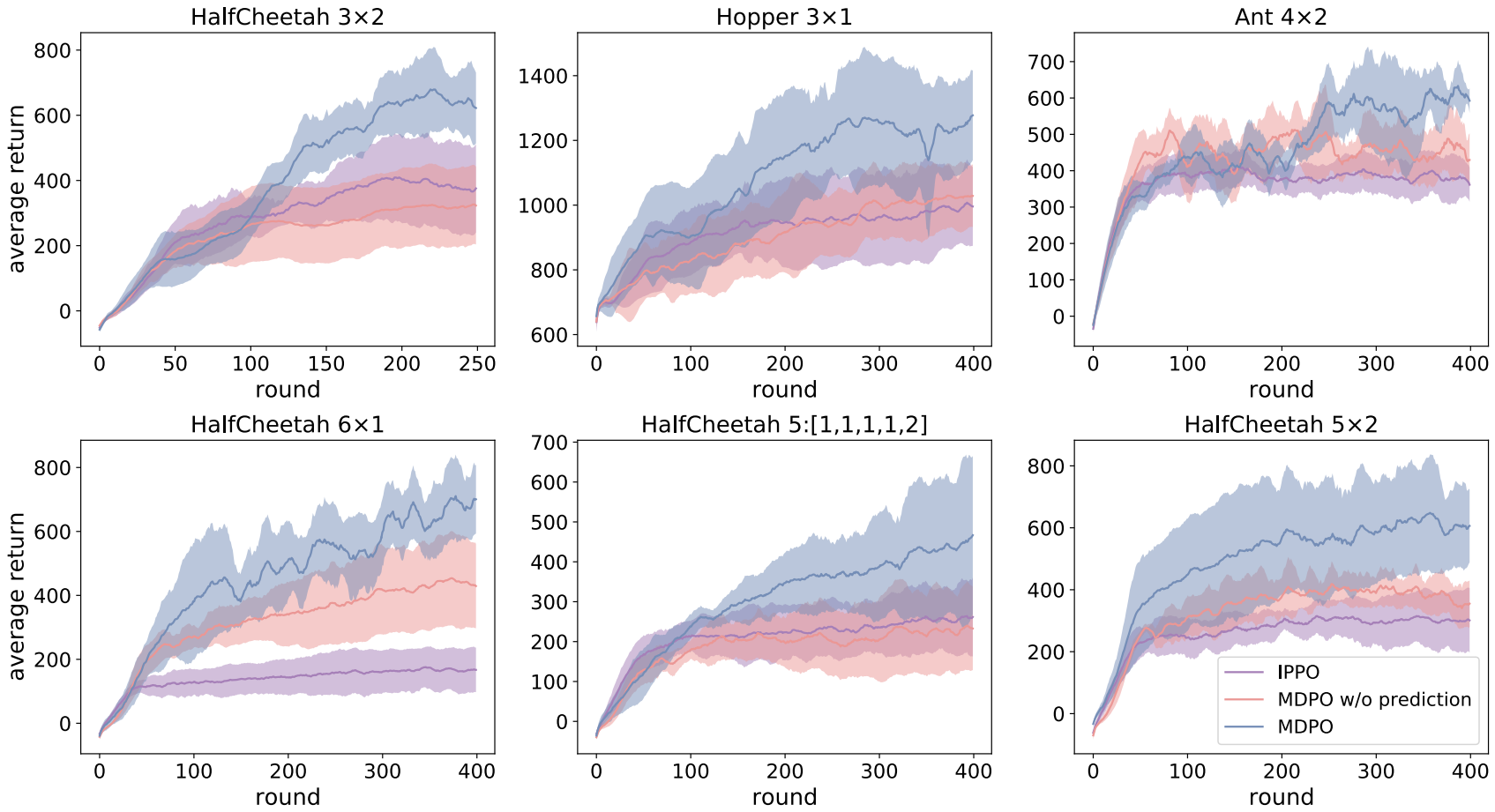
✔ **MDPO indeed induces more stable visitation frequency**

✔ **Latent variable prediction better controls $\hat{\epsilon}_\omega$**

Luo, Jiang, and Lu, Model-based decentralized policy optimization, 2022

MDPO, MDPO w/o Prediction, and IPPO

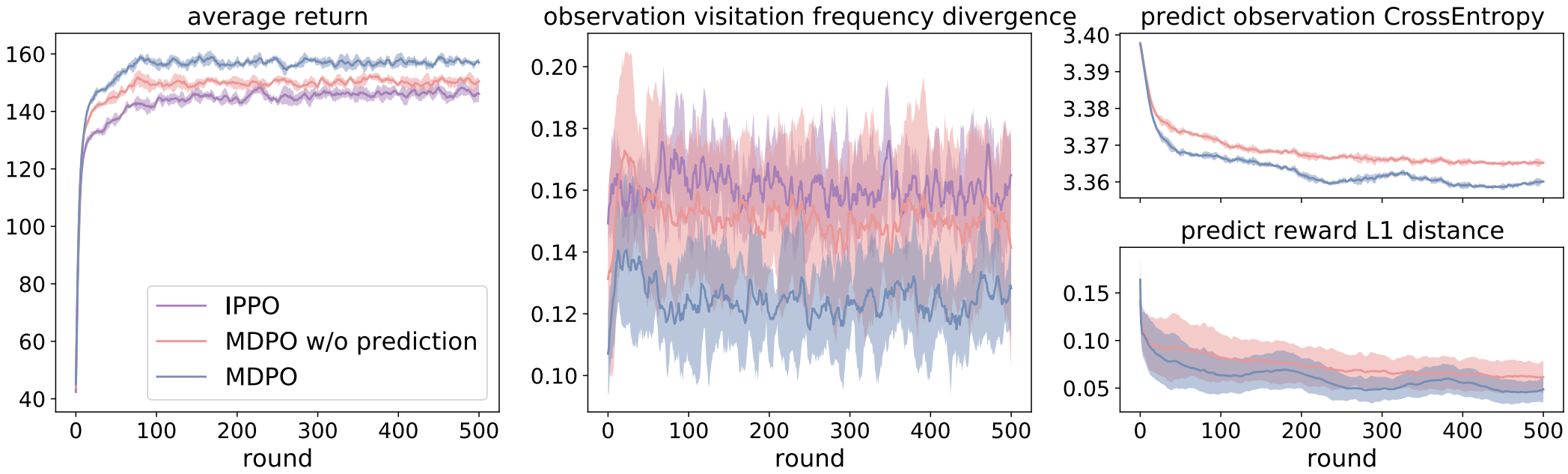
- Muti-agent MuJoCo



Luo, Jiang, and Lu, Model-based decentralized policy optimization, 2022

MDPO, MDPO w/o Prediction, and IPPO

- Does MDPO help in single-agent non-stationary environments?



✔ **MDPO also helps in single-agent non-stationary environments**

Luo, Jiang, and Lu, Model-based decentralized policy optimization, 2022

Outline

1. Fully decentralized Q-learning
2. Fully decentralized policy optimization
3. **???!!!**

✓ Su et al., A Minimalist Approach to Decentralized Multi-Agent Reinforcement Learning, 2022
✓ Jiang and Lu, I2Q: A fully decentralized Q-learning algorithm, 2022
✓ Su and Lu, Fully decentralized policy optimization, 2022
✓ Luo, Jiang, and Lu, Model-based decentralized policy optimization, 2022

Takeaways

✔ Fully decentralized MARL is widely applicable

✔ Fully decentralized MARL can be much better than IQL and IPPO

✘ These methods are preliminary

✘ More efforts are needed

Fully decentralized learning can be a unified paradigm for RL and cooperative MARL!

That is all!
Questions?

zongqing.lu@pku.edu.cn

北京大學
PEKING UNIVERSITY