Trust Region Policy Optimisation in Multi-Agent Reinforcement Learning

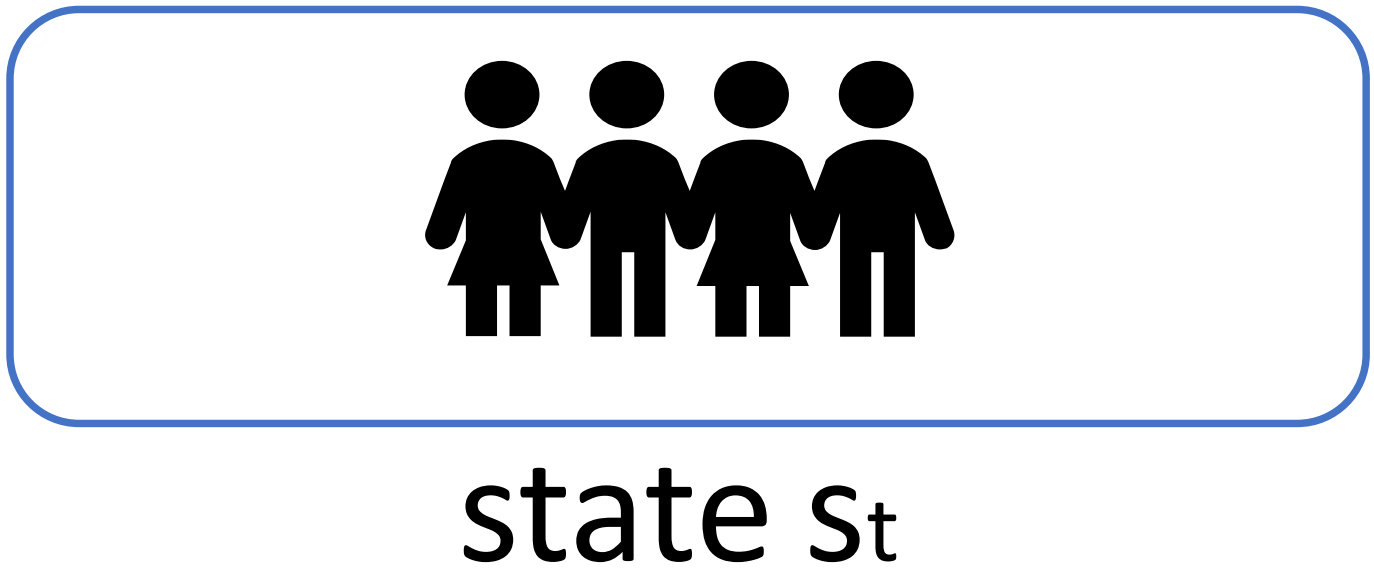
Jakub Grudzien Kuba, Ruiqing Chen,
Muning Wen, Ying Wen, Fanglei Sung, Jun Wang, Yaodong Yang

ICLR 2022

Cooperative MARL: Problem Formulation

Problem Formulation

At time step t, n agents are at state s_t



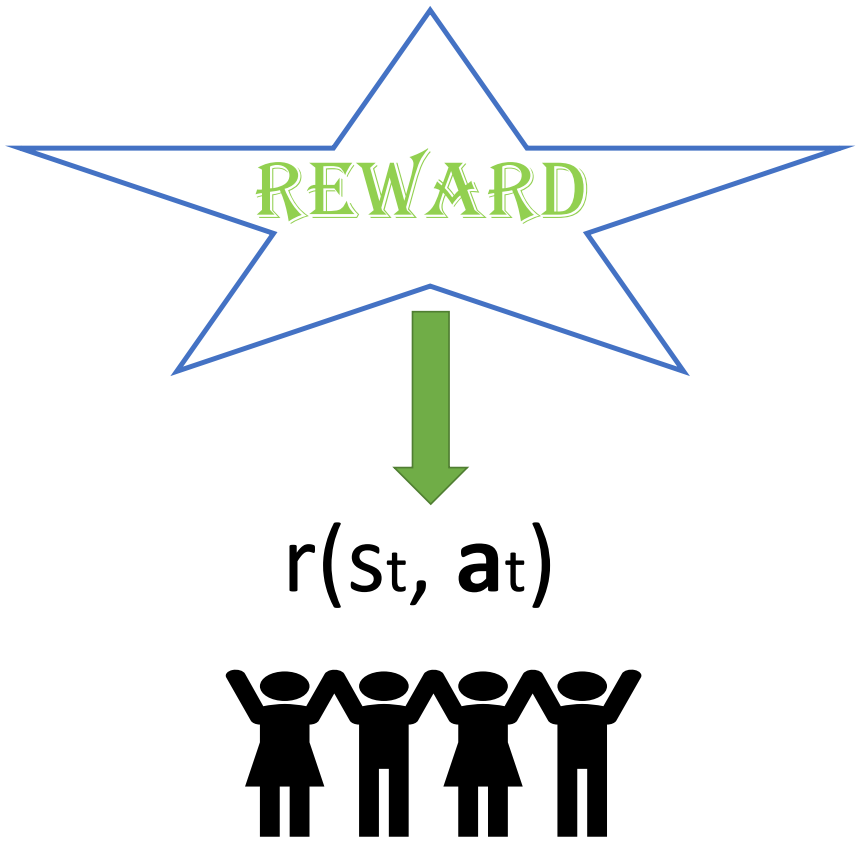
Problem Formulation

The agents take actions $a_t^1 \sim \pi^1(\cdot^1|s_t), \ldots, a_t^n \sim \pi^n(\cdot^n|s_t)$

Equivalently $\mathbf{a}_t \sim \boldsymbol{\pi}(\cdot|s_t)$

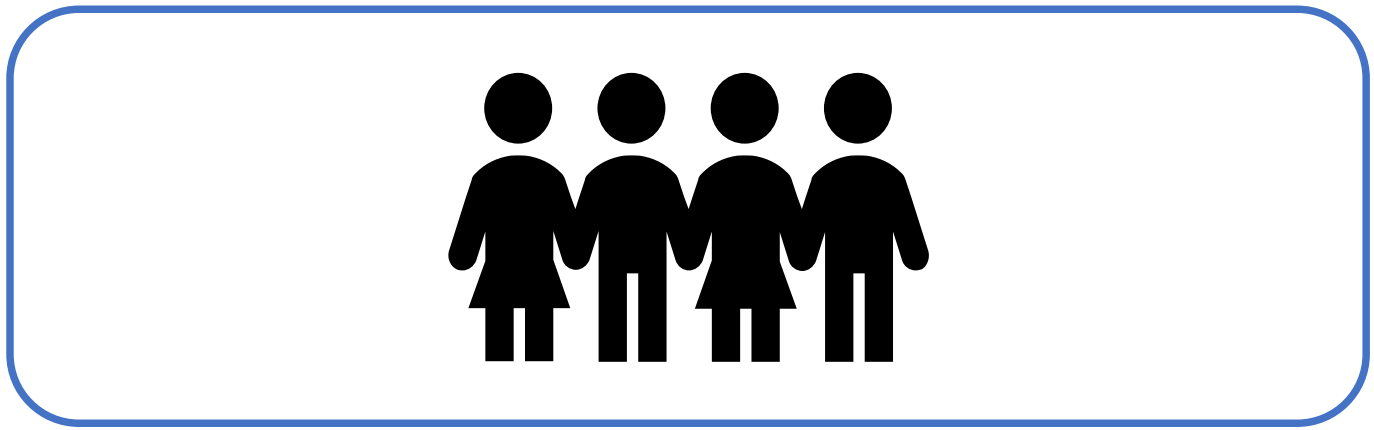
Problem Formulation

The environment emits the joint reward $r(\mathbf{s}_t, \mathbf{a}_t)$



Problem Formulation

The agents move to the next state
$s_{t+1} \sim P(\cdot|s_t, \mathbf{a}_t)$



state s_{t+1}

Problem Formulation

The agents want to maximise the joint return

$$J(\boldsymbol{\pi}) = \mathbb{E}_{\mathbf{s}_0 \sim d^0, a_{0:\infty} \sim \boldsymbol{\pi}, \mathbf{s}_{1:\infty} \sim P}\Big[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)\Big]$$

Multi-Agent Trust Region Learning: A Summary of Existing Approaches

Existing Approaches

Methodology

Current Approaches

Methodology

- To endow all agents with a single policy (**homogeneity**),

$$\pi^i = \pi^j = \pi, \ \ \forall i,j \in \mathcal{N}.$$

Existing Approaches

Methodology

- To endow all agents with a single policy (**homogeneity**),

$$\pi^i = \pi^j = \pi, \ \forall i, j \in \mathcal{N}.$$

- To perform a trust-region update on π, for example, PPO

$$\mathbb{E}_{\mathrm{s}\sim\rho_\pi,\mathrm{a}\sim\pi}\left[\min\left(\frac{\bar{\pi}(\mathrm{a}|\mathrm{s})}{\pi(\mathrm{a}|\mathrm{s})}A_\pi(\mathrm{s},\mathrm{a}), \mathrm{clip}\left(\frac{\bar{\pi}(\mathrm{a}|\mathrm{s})}{\pi(\mathrm{a}|\mathrm{s})}, 1\pm\epsilon\right)A_\pi(\mathrm{s},\mathrm{a})\right)\right].$$

Existing Approaches

Shortcomings

- Suboptimality (possibly exponential) of the solution in case of homogeneous policies

$$\frac{J^*_{\text{share}}}{J^*} = \frac{2}{2^n}.$$

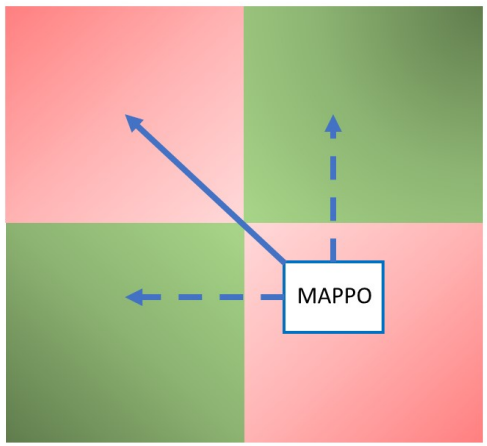
Current Approaches

Shortcomings

- Suboptimality (possibly exponential) of the solution in case of homogeneous policies

$$\frac{J^*_{\text{share}}}{J^*} = \frac{2}{2^n}.$$

- A loss of the monotonic improvement property for **heterogeneous** policies.



Heterogenous-Agent Trust Region Algorithms

Heterogeneous-Agent Trust Region Algorithms

Let $i_{1:m}$ and $j_{1:k}$ be disjoint, ordered subsets of agents.

Heterogeneous-Agent Trust Region Algorithms

Let $i_{1:m}$ and $j_{1:k}$ be disjoint, ordered subsets of agents.

- *The multi-agent state-action velue function*

$$Q_{\boldsymbol{\pi}}^{i_{1:m}}(s, \boldsymbol{a}^{i_{1:m}}) \triangleq \mathbb{E}_{\mathbf{a}^{-i_{1:m}} \sim \boldsymbol{\pi}^{-i_{1:m}}}\left[Q_{\boldsymbol{\pi}}(s, \boldsymbol{a}^{i_{1:m}}, \mathbf{a}^{-i_{1:m}})\right],$$

Heterogeneous-Agent Trust Region Algorithms

Let $i_{1:m}$ and $j_{1:k}$ be disjoint, ordered subsets of agents.

- *The multi-agent state-action velue function*

$$Q_{\boldsymbol{\pi}}^{i_{1:m}}(s, \boldsymbol{a}^{i_{1:m}}) \triangleq \mathbb{E}_{\mathbf{a}^{-i_{1:m}} \sim \boldsymbol{\pi}^{-i_{1:m}}}\left[Q_{\boldsymbol{\pi}}(s, \boldsymbol{a}^{i_{1:m}}, \mathbf{a}^{-i_{1:m}})\right],$$

- *The multi-agent advantage function*

$$A_{\boldsymbol{\pi}}^{i_{1:m}}(s, \boldsymbol{a}^{j_{1:k}}, \boldsymbol{a}^{i_{1:m}}) \triangleq Q_{\boldsymbol{\pi}}^{j_{1:k}, i_{1:m}}(s, \boldsymbol{a}^{j_{1:k}}, \boldsymbol{a}^{i_{1:m}}) - Q_{\boldsymbol{\pi}}^{j_{1:k}}(s, \boldsymbol{a}^{j_{1:k}}).$$

Heterogeneous-Agent Trust Region Algorithms

Let $i_{1:m}$ and $j_{1:k}$ be disjoint, ordered subsets of agents.

- *The multi-agent state-action velue function*

$$Q_{\boldsymbol{\pi}}^{i_{1:m}}(s, \boldsymbol{a}^{i_{1:m}}) \triangleq \mathbb{E}_{\mathbf{a}^{-i_{1:m}} \sim \boldsymbol{\pi}^{-i_{1:m}}}\left[Q_{\boldsymbol{\pi}}(s, \boldsymbol{a}^{i_{1:m}}, \mathbf{a}^{-i_{1:m}})\right],$$

- *The multi-agent advantage function*

$$A_{\boldsymbol{\pi}}^{i_{1:m}}(s, \boldsymbol{a}^{j_{1:k}}, \boldsymbol{a}^{i_{1:m}}) \triangleq Q_{\boldsymbol{\pi}}^{j_{1:k}, i_{1:m}}(s, \boldsymbol{a}^{j_{1:k}}, \boldsymbol{a}^{i_{1:m}}) - Q_{\boldsymbol{\pi}}^{j_{1:k}}(s, \boldsymbol{a}^{j_{1:k}}).$$

Lemma. For any state $s \in \mathcal{S}$, joint action $\boldsymbol{a}^{i_{1:n}} \in \mathcal{A}$, and permutation of agents $i_{1:n}$,

$$A_{\boldsymbol{\pi}}^{i_{1:n}}(s, \boldsymbol{a}^{i_{1:n}}) = \sum_{m=1}^{n} A_{\boldsymbol{\pi}}^{i_m}(s, \boldsymbol{a}^{i_{1:m-1}}, a^{i_m}).$$

Heterogeneous-Agent Trust Region Algorithms

Algorithm 1

Heterogeneous-Agent Trust Region Algorithms

Algorithm 1

- Initialise a joint policy π_0 at random.

Heterogeneous-Agent Trust Region Algorithms

Algorithm 1

- Initialise a joint policy π_0 at random.
- **for** $k = 0, 1, 2, \ldots$

Heterogeneous-Agent Trust Region Algorithms

Algorithm 1

- Initialise a joint policy π_0 at random.
- **for** $k = 0, 1, 2, \ldots$
- Compute $C = \frac{4\gamma \max_{s,\boldsymbol{a}} |A_{\boldsymbol{\pi}}(s,\boldsymbol{a})|}{(1-\gamma)^2}$.

Heterogeneous-Agent Trust Region Algorithms

Algorithm 1

- Initialise a joint policy π_0 at random.
- **for** $k = 0, 1, 2, \ldots$
- Compute $C = \frac{4\gamma \max_{s,\boldsymbol{a}} |A_{\boldsymbol{\pi}}(s,\boldsymbol{a})|}{(1-\gamma)^2}$.
- Draw a random permutation of agents $i_{1:n}$.

Heterogeneous-Agent Trust Region Algorithms

Algorithm 1

- Initialise a joint policy π_0 at random.
- **for** $k = 0, 1, 2, \ldots$
- Compute $C = \frac{4\gamma \max_{s,\boldsymbol{a}} |A_{\boldsymbol{\pi}}(s,\boldsymbol{a})|}{(1-\gamma)^2}$.
- Draw a random permutation of agents $i_{1:n}$.
- **for** $m = 1, \ldots, n$

Heterogeneous-Agent Trust Region Algorithms

Algorithm 1

- Initialise a joint policy π_0 at random.
- **for** $k = 0, 1, 2, \ldots$
- Compute $C = \frac{4\gamma \max_{s,\boldsymbol{a}} |A_{\boldsymbol{\pi}}(s,\boldsymbol{a})|}{(1-\gamma)^2}$.
- Draw a random permutation of agents $i_{1:n}$.
- **for** $m = 1, \ldots, n$

$$\pi_{k+1}^{i_m} = \arg\max_{\pi^{i_m}} \mathbb{E}_{\mathrm{s}\sim\rho_{\boldsymbol{\pi}_k}, \mathbf{a}^{i_{1:m-1}}\sim\boldsymbol{\pi}_{k+1}^{i_{1:m-1}}, \mathrm{a}^{i_m}\sim\pi^{i_m}} \left[A_{\boldsymbol{\pi}_k}^{i_m}(s, \mathbf{a}^{i_{1:m-1}}, \mathrm{a}^{i_m})\right] - C\mathrm{D}_{\mathrm{KL}}^{\max}(\pi_k^{i_m}, \pi^{i_m}).$$

Heterogeneous-Agent Trust Region Algorithms

Algorithm 1

- Initialise a joint policy $\boldsymbol{\pi}_0$ at random.
- **for** $k = 0, 1, 2, \ldots$
- Compute $C = \frac{4\gamma \max_{s,\boldsymbol{a}} |A_{\boldsymbol{\pi}}(s,\boldsymbol{a})|}{(1-\gamma)^2}$.
- Draw a random permutation of agents $i_{1:n}$.
- **for** $m = 1, \ldots, n$

$$\pi_{k+1}^{i_m} = \arg\max_{\pi^{i_m}} \mathbb{E}_{s\sim\rho_{\boldsymbol{\pi}_k}, \mathbf{a}^{i_{1:m-1}}\sim\boldsymbol{\pi}_{k+1}^{i_{1:m-1}}, a^{i_m}\sim\pi^{i_m}} \left[A_{\boldsymbol{\pi}_k}^{i_m}(s, \mathbf{a}^{i_{1:m-1}}, a^{i_m})\right] - C D_{\mathrm{KL}}^{\max}(\pi_k^{i_m}, \pi^{i_m}).$$

Theorem. Algorithm 1 achieves the monotonic improvement property, $J(\boldsymbol{\pi}_{k+1}) \geq J(\boldsymbol{\pi}_k)$, and converges to a set of Nash equilibria.

Deep Heterogeneous-Agent Trust Region Algorithms

Deep Heterogeneous-Agent Trust Region Algorithms

Update the parameters to $\theta_{k+1}^{i_m}$ by maximising

Deep Heterogeneous-Agent Trust Region Algorithms

Update the parameters to $\theta^{i_m}_{k+1}$ by maximising

HATRPO

$$\mathbb{E}_{\mathrm{s}\sim\rho_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}},\mathbf{a}^{i_{1:m-1}}\sim\boldsymbol{\pi}^{i_{1:m-1}}_{\boldsymbol{\theta}^{i_{1:m-1}}_{k+1}},\mathrm{a}^{i_m}\sim\pi^{i_m}_{\theta^{i_m}}}\left[A^{i_m}_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}(\mathrm{s},\mathbf{a}^{i_{1:m-1}},\mathrm{a}^{i_m})\right],$$

$$\text{subject to } \mathbb{E}_{\mathrm{s}\sim\rho_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}}\left[\mathrm{D}_{\mathrm{KL}}\left(\pi^{i_m}_{\theta^{i_m}_k}(\cdot|\mathrm{s}),\pi^{i_m}_{\theta^{i_m}}(\cdot|\mathrm{s})\right)\right]\leq\delta.$$

Deep Heterogeneous-Agent Trust Region Algorithms

Update the parameters to $\theta_{k+1}^{i_m}$ by maximising

HATRPO

$$\mathbb{E}_{\mathrm{s}\sim\rho_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}},\mathbf{a}^{i_{1:m-1}}\sim\boldsymbol{\pi}_{\boldsymbol{\theta}_{k+1}}^{i_{1:m-1}},\mathrm{a}^{i_m}\sim\pi_{\theta^{i_m}}^{i_m}}\left[A_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}^{i_m}(\mathrm{s},\mathbf{a}^{i_{1:m-1}},\mathrm{a}^{i_m})\right],$$

$$\text{subject to } \mathbb{E}_{\mathrm{s}\sim\rho_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}}\left[\mathrm{D}_{\mathrm{KL}}\left(\pi_{\theta_k^{i_m}}^{i_m}(\cdot|\mathrm{s}),\pi_{\theta^{i_m}}^{i_m}(\cdot|\mathrm{s})\right)\right]\leq\delta.$$

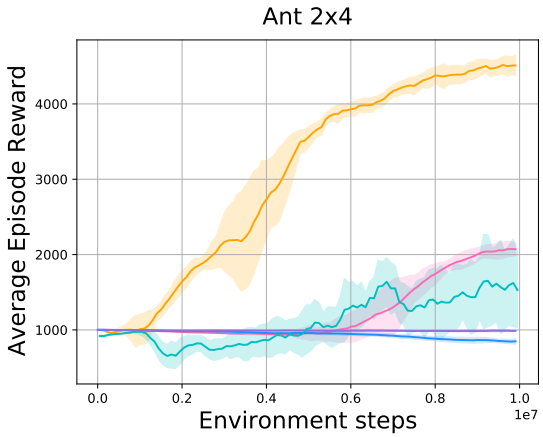
HAPPO

$$\mathbb{E}_{\mathrm{s}\sim\rho_{\boldsymbol{\pi}_{\boldsymbol{\theta}_k}},\mathbf{a}\sim\boldsymbol{\pi}_{\boldsymbol{\theta}_k}}\left[\min\left(\frac{\pi_{\theta^{i_m}}^{i_m}(\mathrm{a}^i|\mathrm{s})}{\pi_{\theta_k^{i_m}}^{i_m}(\mathrm{a}^i|\mathrm{s})}M^{i_{1:m}}(\mathrm{s},\mathbf{a}),\mathrm{clip}\left(\frac{\pi_{\theta^{i_m}}^{i_m}(\mathrm{a}^i|\mathrm{s})}{\pi_{\theta_k^{i_m}}^{i_m}(\mathrm{a}^i|\mathrm{s})},1\pm\epsilon\right)M^{i_{1:m}}(\mathrm{s},\mathbf{a})\right)\right],$$

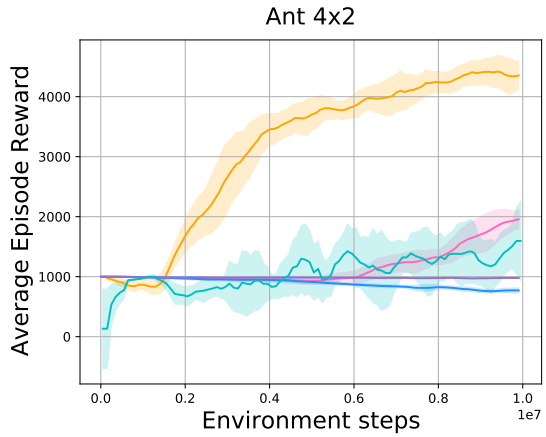
$$\text{where } M^{i_{1:m}}(\mathrm{s},\mathbf{a})=\frac{\pi_{\boldsymbol{\theta}_{k+1}}^{i_{1:m-1}}(\mathbf{a}^{i_{1:m-1}}|\mathrm{s})}{\pi_{\boldsymbol{\theta}_k^{i_{1:m-1}}}^{i_{1:m-1}}(\mathbf{a}^{i_{1:m-1}}|\mathrm{s})}A_{\boldsymbol{\pi}}(\mathrm{s},\mathbf{a}).$$

Empirical Results: the New State of the Art

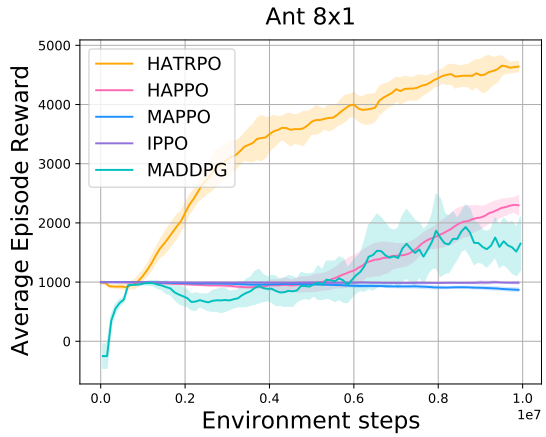
Multi-Agent MuJoCo



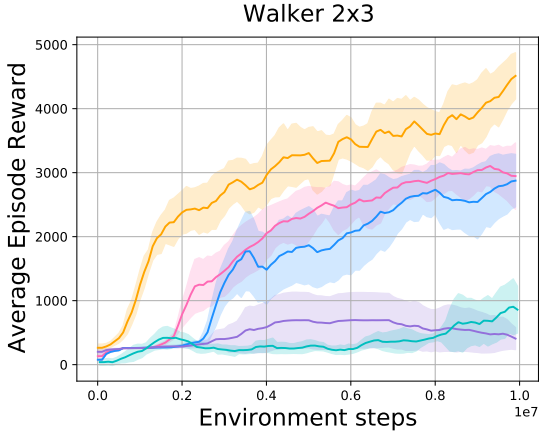
(a) 2x4-Agent Ant



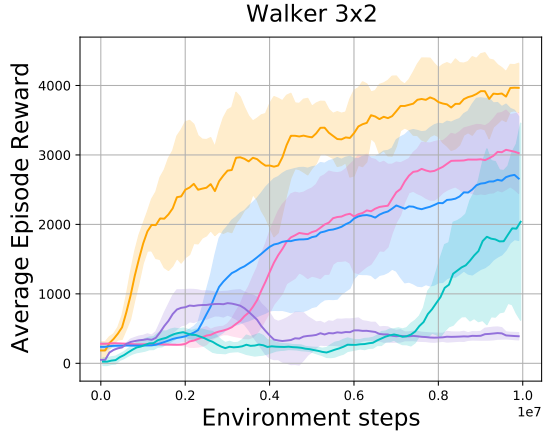
(b) 4x2-Agent Ant



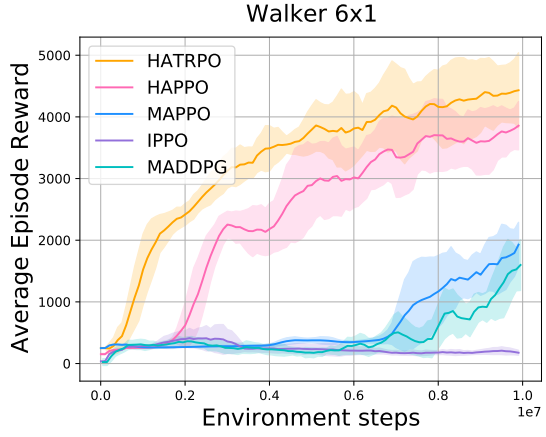
(c) 8x1-Agent Ant



(d) 2x3-Agent Walker



(e) 3x2-Agent Walker



(f) 6x1-Agent Walker

Thank you for your attention!

- *Jakub Grudzien Kuba* (University of Oxford, Huawei Technologies)
- *Ruiqing Chen* (ShanghaiTech University)
- *Muning Wen* (Shanghai Jiao Tong University)
- *Ying Wen* (Shanghai Jiao Tong University)
- *Fanglei Sun* (ShanghaiTech University)
- *Jun Wang* (University College London)
- *Yaodong Yang* (Peking University)