Order Matters: Agent-by-agent Policy Optimization

Xihuai Wang [1,2] **Zheng Tian** [3] **Ziyu Wan** [1,2] **Ying Wen** [1] **Jun Wang** [2,4] **Weinan Zhang** [1]

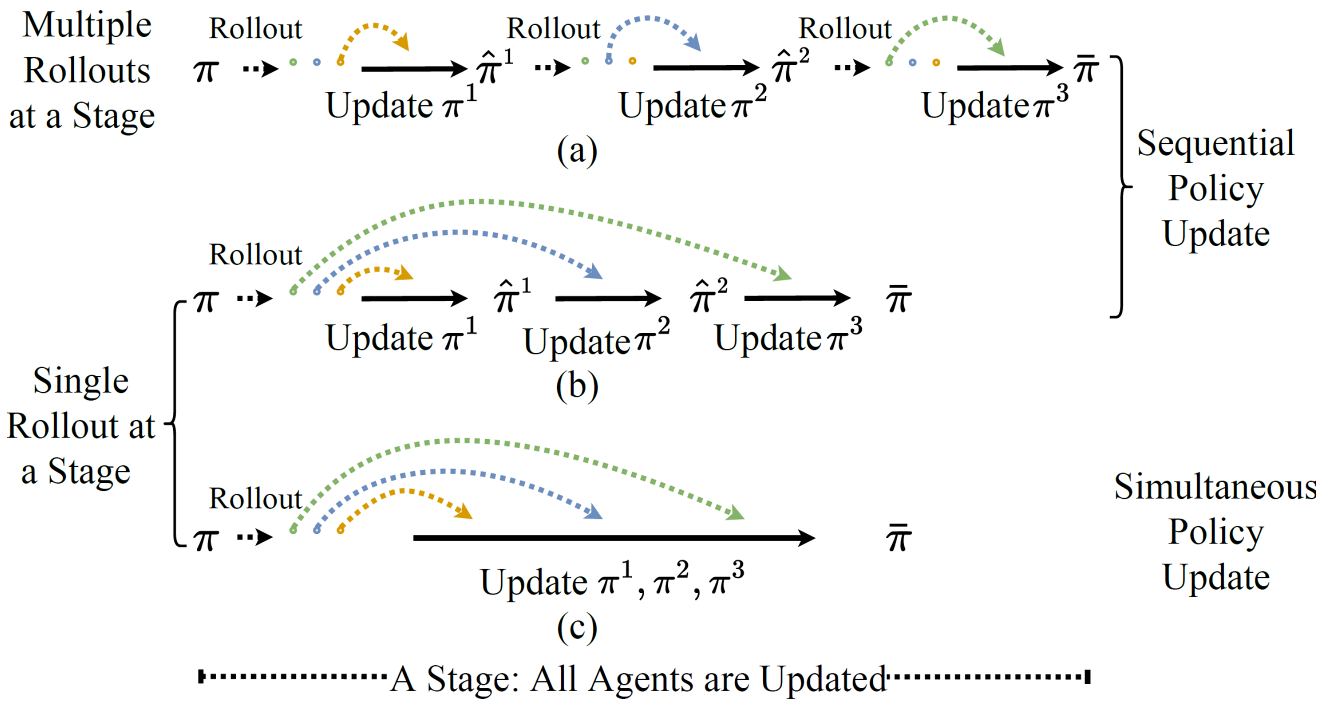
[1] Shanghai Jiao Tong University [2] Digital Brain Lab [3] ShanghaiTech University [4] University College London

Rollout Scheme and Policy Update Scheme

- Several works that adopt trust region learning in multi-agent reinforcement learning (MARL) have been proposed.
- Most algorithms update the agents simultaneously, that is, all agents perform policy improvement at the same time and cannot observe the change of other agents.
- The simultaneous update scheme brings about the non-stationarity problem, i.e., the environment dynamic changes from one agent's perspective as other agents also change their policies.

Rollout Scheme and Policy Update Scheme

- Algorithms that sequentially execute agent-by agent updates allow agents to perceive changes made by preceding agents, presenting another perspective for analyzing inter-agent interaction.
- Alleviate the problems brought by simultaneous update scheme.
- Algorithms in sequential policy update scheme can be further categorized by whether a rollout is sampled after an agent's policy is updated.



Sequential Policy Update Scheme

- We formulate the update process in sequential policy update scheme as:

$$\boldsymbol{\pi} = \hat{\boldsymbol{\pi}}^0 \xrightarrow[\text{Update } \pi^1]{\max_{\pi^1} \mathcal{L}_{\boldsymbol{\pi}}(\hat{\boldsymbol{\pi}}^1)} \hat{\boldsymbol{\pi}}^1 \rightarrow \cdots \rightarrow \hat{\boldsymbol{\pi}}^{n-1} \xrightarrow[\text{Update } \pi^n]{\max_{\pi^n} \mathcal{L}_{\hat{\boldsymbol{\pi}}^{n-1}}(\hat{\boldsymbol{\pi}}^n)} \hat{\boldsymbol{\pi}}^n = \bar{\boldsymbol{\pi}}.$$

Naive Sequential Policy Updating with Single Rollout Fails

- An intuitive surrogate objective of agent i can be designed directly following the construction of surrogate objective in TRPO:

$$\mathcal{L}^{I}_{\hat{\boldsymbol{\pi}}^{i-1}}(\hat{\boldsymbol{\pi}}^{i}) = \mathcal{J}(\hat{\boldsymbol{\pi}}^{i-1}) + \frac{1}{1-\gamma}\mathbb{E}_{(s,\boldsymbol{a})\sim(d^{\boldsymbol{\pi}},\hat{\boldsymbol{\pi}}^{i})}[A^{\boldsymbol{\pi}}(s,\boldsymbol{a})]$$

Proposition 1 *For agent i, let* $\epsilon = \max_{s,\boldsymbol{a}} |A^{\boldsymbol{\pi}}(s,\boldsymbol{a})|$, $\alpha^j = D^{\max}_{TV}(\pi^j \| \bar{\pi}^j)\ \forall j \in (e^i \cup \{i\})$, *where* $D_{TV}(p\|q)$ *is the total variation distance between distributions p and q and we define* $D^{\max}_{TV}(\pi\|\bar{\pi}) = \max_s D_{TV}(\pi(\cdot|s)\|\bar{\pi}(\cdot|s))$, *then we have:*

$$\left|\mathcal{J}(\hat{\boldsymbol{\pi}}^{i}) - \mathcal{L}^{I}_{\hat{\boldsymbol{\pi}}^{i-1}}(\hat{\boldsymbol{\pi}}^{i})\right| \le 2\epsilon\alpha^i\left(\frac{3}{1-\gamma} - \frac{2}{1-\gamma(1-\sum_{j\in(e^i\cup\{i\})}\alpha^j)}\right) + \overbrace{\frac{2\epsilon\sum_{j\in e^i}\alpha^j}{1-\gamma}}^{\textit{Uncontrollable}} = \beta^I_i$$

The uncontrollable term results in that the performance of the future joint policy $\hat{\boldsymbol{\pi}}^{i}$ **may not be improved even if** $\boldsymbol{\alpha^i}$ **is well constrained**.

Preceding-agent Off-policy Correction

- The uncontrollable term is caused by one ignoring how the updating of its preceding agents' policies influences its advantage function. We investigate reducing the uncontrollable term in policy evaluation.
- Preceding-agent Off-policy Correction (PreOPC):

$$A^{\boldsymbol{\pi},\hat{\boldsymbol{\pi}}^{i-1}}(s_t,\boldsymbol{a}_t)=\delta_t+\sum_{k\geq 1}\gamma^k\Big(\prod_{j=1}^{k}\lambda\min\big(1.0,\frac{\hat{\boldsymbol{\pi}}^{i-1}(\boldsymbol{a}_{t+j}|s_{t+j})}{\boldsymbol{\pi}(\boldsymbol{a}_{t+j}|s_{t+j})}\big)\Big)\delta_{t+k}$$

$$\delta_t=r(s_t,\boldsymbol{a}_t)+\gamma V(s_{t+1})-V(s_t)$$

- We prove that $A^{\boldsymbol{\pi},\hat{\boldsymbol{\pi}}^{i-1}}$ approximates $A^{\hat{\boldsymbol{\pi}}^{i-1}}$ as the agent i update its value function.

Tighter Monotonic Improvement Bound

- With PreOPC, the surrogate objective of agent i becomes:

$$\mathcal{L}_{\hat{\boldsymbol{\pi}}^{i-1}}(\hat{\boldsymbol{\pi}}^{i}) = \mathcal{J}(\hat{\boldsymbol{\pi}}^{i-1}) + \frac{1}{1-\gamma}\mathbb{E}_{(s,\boldsymbol{a})\sim(d^{\boldsymbol{\pi}},\hat{\boldsymbol{\pi}}^{i})}[A^{\boldsymbol{\pi},\hat{\boldsymbol{\pi}}^{i-1}}(s,\boldsymbol{a})]$$

Theorem 1 (Single Agent Monotonic Bound) *For agent i, let* $\epsilon^i = \max_{s,\boldsymbol{a}} |A^{\hat{\boldsymbol{\pi}}^{i-1}}(s,\boldsymbol{a})|$, $\xi^i = \max_{s,\boldsymbol{a}} |A^{\boldsymbol{\pi},\hat{\boldsymbol{\pi}}^{i-1}}(s,\boldsymbol{a}) - A^{\hat{\boldsymbol{\pi}}^{i-1}}(s,\boldsymbol{a})|$, $\alpha^j = D_{TV}^{\max}(\pi^j \| \bar{\pi}^j)\ \forall j \in (e^i \cup \{i\})$, *then we have:*

$$\left|\mathcal{J}(\hat{\boldsymbol{\pi}}^{i}) - \mathcal{L}_{\hat{\boldsymbol{\pi}}^{i-1}}(\hat{\boldsymbol{\pi}}^{i})\right| \le 4\epsilon^i\alpha^i\left(\frac{1}{1-\gamma} - \frac{1}{1-\gamma(1-\sum_{j\in(e^i\cup\{i\})}\alpha^j)}\right) + \frac{\xi^i}{1-\gamma}$$

$$\le \frac{4\gamma\epsilon^i}{(1-\gamma)^2}\left(\alpha^i \sum_{j\in(e^i\cup\{i\})} \alpha^j\right) + \frac{\xi^i}{1-\gamma}.$$

Tighter Monotonic Improvement Bound

Table 1: Comparisons of trust region MARL algorithms. The proofs of the monotonic bounds can be found in Appx. A. Note that we also provide the monotonic bound of RPISA-PPO, which implements RPISA with PPO as the base algorithm. We separate RPISA-PPO from other methods as it has low sample efficiency and thus does not constitute a fair comparison.

Algorithm	Rollout	Update	Sample Efficiency	Monotonic Bound
RPISA-PPO	Multiple	Sequential	Low	$4\epsilon\sum_{i=1}^{n}\alpha^i(\frac{1}{1-\gamma}-\frac{1}{1-\gamma(1-\alpha^i)})$ Single Agent: $4\epsilon\alpha^i(\frac{1}{1-\gamma}-\frac{1}{1-\gamma(1-\alpha^i)})$
MAPPO	Single	Simultaneous	High	$4\epsilon\sum_{i=1}^{n}\frac{\alpha^i}{1-\gamma}$
CoPPO	Single	Simultaneous	High	$4\epsilon\sum_{i=1}^{n}\alpha^i(\frac{1}{1-\gamma}-\frac{1}{1-\gamma(1-\sum_{j=1}^{n}\alpha^j)})$
HAPPO	Single	Sequential	High	$4\epsilon\sum_{i=1}^{n}\alpha^i(\frac{1}{1-\gamma}-\frac{1}{1-\gamma(1-\sum_{j=1}^{n}\alpha^j)})$ Single Agent: No Guarantee
A2PO (ours)	Single	Sequential	High	$4\epsilon\sum_{i=1}^{n}\alpha^i(\frac{1}{1-\gamma}-\frac{1}{1-\gamma(1-\sum_{j\in(e^i\cup\{i\})}\alpha^j)})+\frac{\sum_{i=1}^{n}\xi^i}{1-\gamma}$ Single Agent: $4\epsilon^i\alpha^i(\frac{1}{1-\gamma}-\frac{1}{1-\gamma(1-\sum_{j\in(e^i\cup\{i\})}\alpha^j)})+\frac{\xi^i}{1-\gamma}$

- Considering that $\forall\, i$, ξ^i converges to 0, we get tighter monotonic improvement bound compared to previous trust region methods in multi-agent scenarios. **A tighter bound improves expected performance by optimizing the surrogate objective more effectively.**

Agent-by-agent Policy Optimization

- The practical objective of updating agent i becomes:

$$\tilde{\mathcal{L}}_{\hat{\boldsymbol{\pi}}^{i-1}}(\hat{\boldsymbol{\pi}}^i) = \mathbb{E}_{(s,\boldsymbol{a})\sim(d^{\boldsymbol{\pi}},\boldsymbol{\pi})}\left[\min\left(l(s,\boldsymbol{a})A^{\boldsymbol{\pi},\hat{\boldsymbol{\pi}}^{i-1}}, \text{clip}\left(l(s,\boldsymbol{a}), 1\pm\epsilon^i\right)A^{\boldsymbol{\pi},\hat{\boldsymbol{\pi}}^{i-1}}\right)\right]$$

$$\text{where } l(s,\boldsymbol{a}) = \frac{\bar{\pi}^i(a^i|s)}{\pi^i(a^i|s)}g(s,\boldsymbol{a}), \text{ and } g(s,\boldsymbol{a}) = \text{clip}\left(\frac{\prod_{j\in e^i}\bar{\pi}^j(a^j|s)}{\prod_{j\in e^i}\pi^j(a^j|s)}, 1\pm\frac{\epsilon^i}{2}\right)$$

Algorithm 1: Agent-by-agent Policy Optimization (A2PO)

Initialize the joint policy $\boldsymbol{\pi}_0 = \{\pi_0^1, \ldots, \pi_0^n\}$, and the global value function V.
for *iteration* $m = 1, 2, \ldots$ **do**
  Collect data using $\boldsymbol{\pi}_{m-1} = \{\pi_{m-1}^1, \ldots, \pi_{m-1}^n\}$.
  for *Order* $k = 1, \ldots, n$ **do**
    Select an agent according to the selection rule as $i = \mathcal{R}(k)$.
    Policy $\pi_m^i = \pi_{m-1}^i$, preceding agents $e^i = \{\mathcal{R}(1), \ldots, \mathcal{R}(k-1)\}$.
    Joint policy $\hat{\boldsymbol{\pi}}^i = \{\pi_m^i, \pi_m^{j\in e^k}, \pi_{m-1}^{j\in\mathcal{N}-e^k}\}$.
    Compute the advantage approximation as $A^{\boldsymbol{\pi},\hat{\boldsymbol{\pi}}^{i-1}}(s,\boldsymbol{a})$ via Eq. (2).
    Compute the value target $v(s_t) = A^{\boldsymbol{\pi},\hat{\boldsymbol{\pi}}^{i-1}}(s,\boldsymbol{a}) + V(s)$.
    for P *epochs* **do**
      $\pi_m^i = \arg\max_{\pi_m^i} \tilde{\mathcal{L}}_{\hat{\boldsymbol{\pi}}^{i-1}}(\hat{\boldsymbol{\pi}}^i)$ as in Eq. (6).
      $V = \arg\min_V \mathbb{E}_{s\sim d^{\boldsymbol{\pi}}}\|v(s) - V(s)\|^2$.

Agent-by-agent Policy Optimization

- **Semi-greedy Agent Selection Rule**
  - Select the agent to update in order k by

$$\begin{cases} \mathcal{R}(k) = \arg\max_{i \in (\mathcal{N}-e)} \mathbb{E}_{s,a^i}[|A^{\boldsymbol{\pi},\hat{\boldsymbol{\pi}}^{\mathcal{R}(k-1)}}|], & k\%2 = 0 \\ \mathcal{R}(k) \sim \mathcal{U}(\mathcal{N}-e), & k\%2 = 1 \end{cases},$$

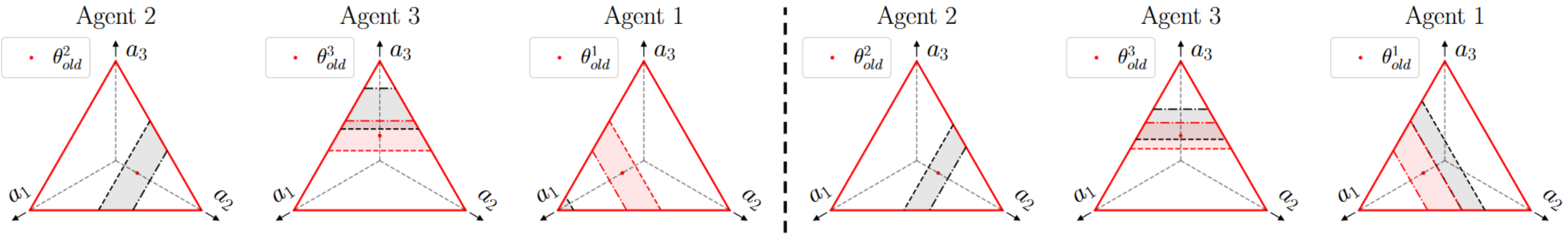
where $e = \{\mathcal{R}(1), \ldots, \mathcal{R}(k-1)\}$

- **Adaptive Clipping Parameter**
  - Adjust the clipping parameters according to the updating order:

$$\mathcal{C}(\epsilon, k) = \epsilon \cdot c_\epsilon + \epsilon \cdot (1 - c_\epsilon) \cdot k/n$$

Algorithm 1: Agent-by-agent Policy Optimization (A2PO)

Initialize the joint policy $\boldsymbol{\pi}_0 = \{\pi_0^1, \ldots, \pi_0^n\}$, and the global value function V.
for *iteration* $m = 1, 2, \ldots$ **do**
  Collect data using $\boldsymbol{\pi}_{m-1} = \{\pi_{m-1}^1, \ldots, \pi_{m-1}^n\}$.
  for *Order* $k = 1, \ldots, n$ **do**
    Select an agent according to the selection rule as $i = \mathcal{R}(k)$.
    Policy $\pi_m^i = \pi_{m-1}^i$, preceding agents $e^i = \{\mathcal{R}(1), \ldots, \mathcal{R}(k-1)\}$.
    Joint policy $\hat{\boldsymbol{\pi}}^i = \{\pi_m^i, \pi_m^{j \in e^k}, \pi_{m-1}^{j \in \mathcal{N}-e^k}\}$.
    Compute the advantage approximation as $A^{\boldsymbol{\pi},\hat{\boldsymbol{\pi}}^{i-1}}(s, \boldsymbol{a})$ via Eq. (2).
    Compute the value target $v(s_t) = A^{\boldsymbol{\pi},\hat{\boldsymbol{\pi}}^{i-1}}(s, \boldsymbol{a}) + V(s)$.
    for P *epochs* **do**
      $\pi_m^i = \arg\max_{\pi_m^i} \tilde{\mathcal{L}}_{\hat{\boldsymbol{\pi}}^{i-1}}(\hat{\boldsymbol{\pi}}^i)$ as in Eq. (6).
      $V = \arg\min_V \mathbb{E}_{s \sim d^{\boldsymbol{\pi}}} \|v(s) - V(s)\|^2$.



Experiments

- StarCraftII Multi-agent Challenge (SMAC)
- Multi-agent MuJoCo (MA-MuJoCo)
- Google Research Football Full-game Scenarios
- Multi-agent Particle Environment
- Ablation Study
- Training Duration

Experiments

- **StarCraftII Multi-agent Challenge (SMAC)**

Table 5: Median win rates and standard deviations on SMAC tasks. ‘w/ PS’ means the algorithm is implemented as parameter sharing

Map	Difficulty	MAPPO w/ PS	CoPPO w/ PS	HAPPO w/ PS	A2PO w/ PS	Qmix w/ PS
MMM	Easy	96.9(0.988)	96.9(1.25)	95.3(2.48)	**100(1.07)**	95.3(2.5)
3s_vs_5z	Hard	**100(1.17)**	**100(2.08)**	**100(0.659)**	**100(0.534)**	98.4(2.4)
2c_vs_64zg	Hard	**98.4(1.74)**	96.9(0.521)	96.9(0.521)	96.9(0.659)	92.2(4.0)
3s5z	Hard	84.4(4.39)	92.2(2.35)	92.2(1.74)	**98.4(1.04)**	88.3(2.9)
5m_vs_6m	Hard	84.4(2.77)	84.4(2.12)	87.5(2.51)	**90.6(3.06)**	75.8(3.7)
8m_vs_9m	Hard	84.4(2.39)	84.4(2.04)	96.9(3.78)	**100(1.04)**	92.2(2.0)
10m_vs_11m	Hard	93.8(18.7)	96.9(2.6)	98.4(2.99)	**100(0.521)**	95.3(1.0)
6h_vs_8z	Super Hard	87.5(1.53)	**90.6(0.765)**	87.5(1.49)	**90.6(1.32)**	9.4(2.0)
3s5z_vs_3s6z	Super Hard	82.8(19.2)	84.4(2.9)	37.5(13.2)	**93.8(19.8)**	82.8(5.3)
MMM2	Super Hard	90.6(8.89)	90.6(6.93)	51.6(9.01)	**98.4(1.25)**	87.5(2.6)
27m_vs_30m	Super Hard	93.8(3.75)	93.8(2.2)	90.6(4.77)	**100(1.55)**	39.1(9.8)
corridor	Super Hard	96.9(0)	**100(0.659)**	96.9(0.96)	**100(0)**	84.4(2.5)
overall	/	91.1(5.46)	92.6(2.2)	85.9(3.68)	**97.4(2.65)**	78.4(3.6)

Experiments

- **Multi-agent MuJoCo (MA-MuJoCo)**

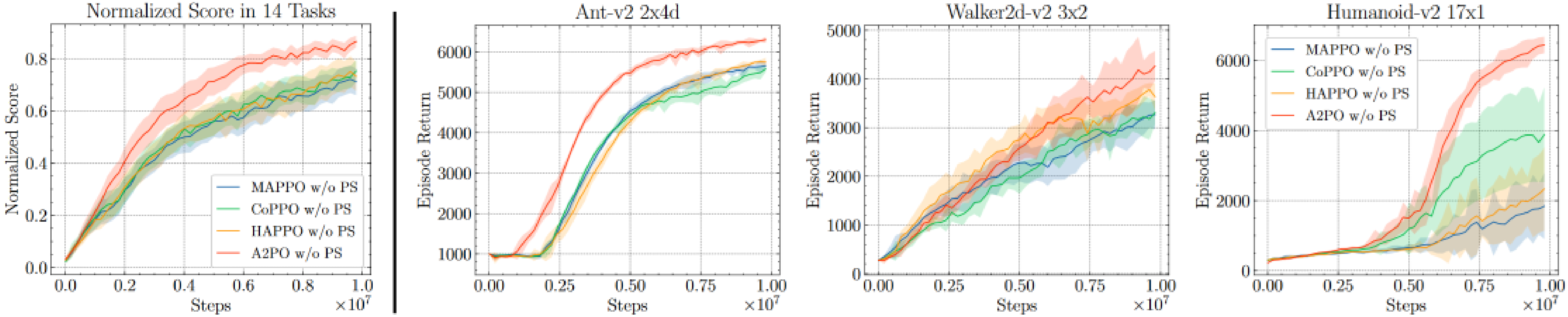


Figure 3: Experiments in MA-MuJoCo. **Left**: Normalized scores on all the 14 tasks. **Right**: Comparisons of averaged return on selected tasks. The number of robot joints increases from left to right.

Experiments

- **Google Research Football Full-game Scenarios**

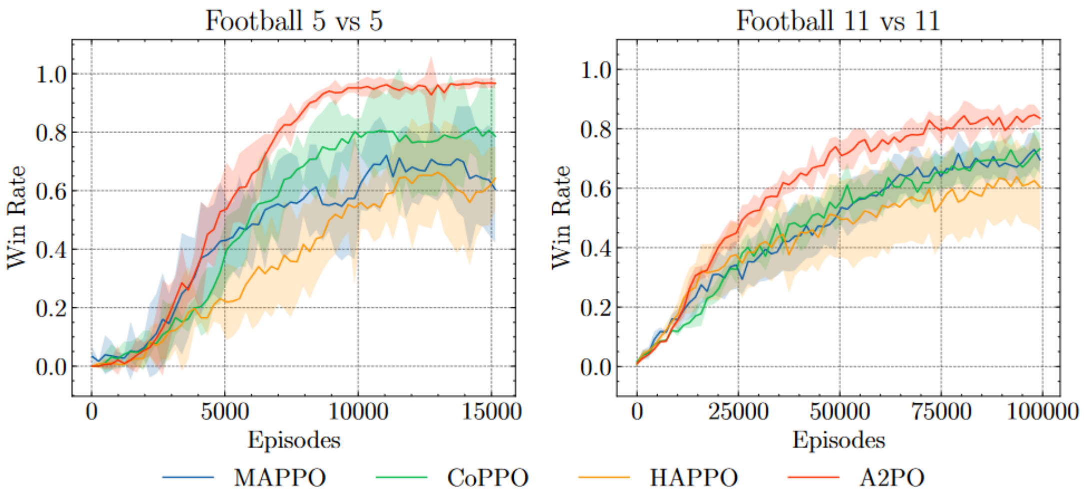


Figure 4: Averaged win rate on the Google Research Football full-game scenarios.

Table 3: Learned behaviors on the Google Research Football 5-vs-5 scenario. Bigger values are better except fot the 'Lost' metric.

Metric	MAPPO	CoPPO	HAPPO	A2PO
Assist	0.04(0.02)	0.19(0.08)	0.07(0.05)	**0.56(0.20)**
Goal	1.95(1.17)	4.42(2.08)	2.68(0.86)	**9.01(0.95)**
Lost	**0.49(0.11)**	0.74(0.33)	1.04(0.12)	0.78(0.15)
Pass	1.52(0.13)	3.44(1.04)	4.03(1.97)	**6.42(2.23)**
Pass Rate	19.3(10.0)	35.0(10.3)	48.9(25.7)	**67.1(11.7)**

Experiments

- **Multi-agent Particle Environment**

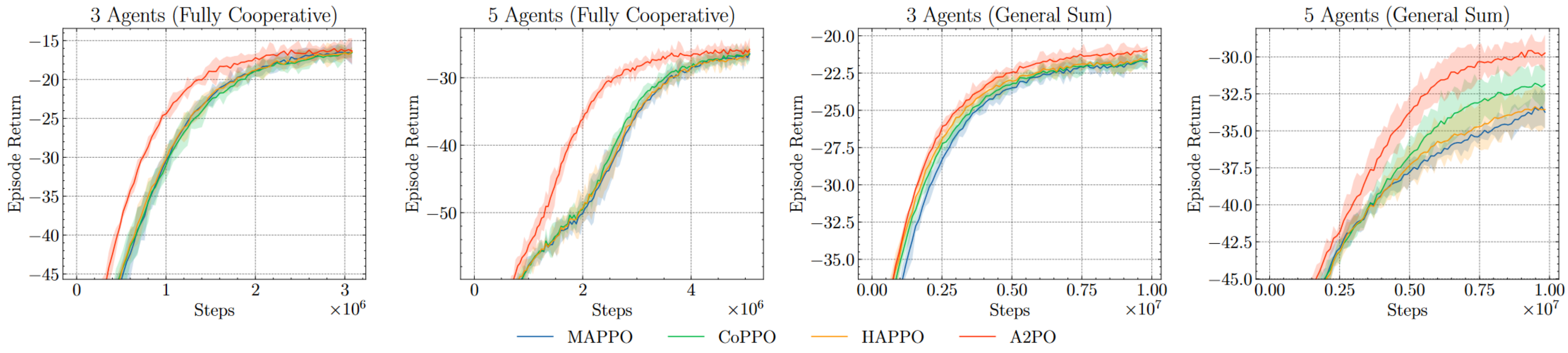
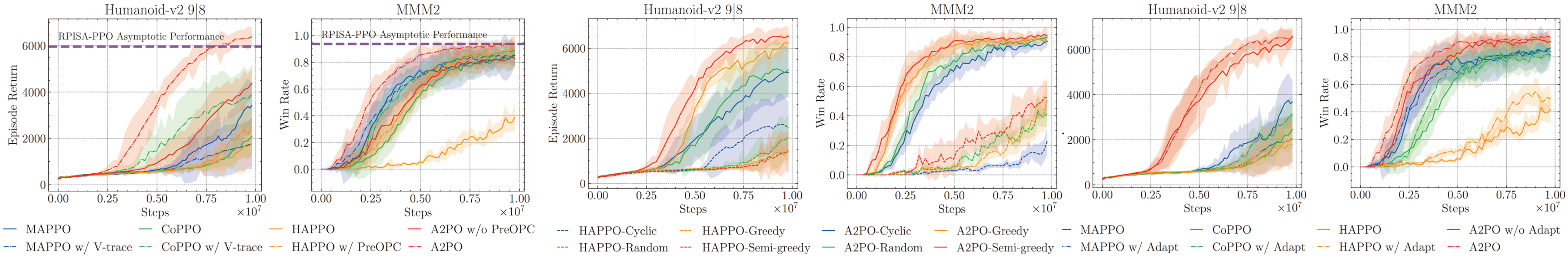


Figure 10: Comparisons of averaged return on the Multi-agent Particle Environment Navigation task. **Left**: The fully cooperative setting. **Right**: The general-sum setting.

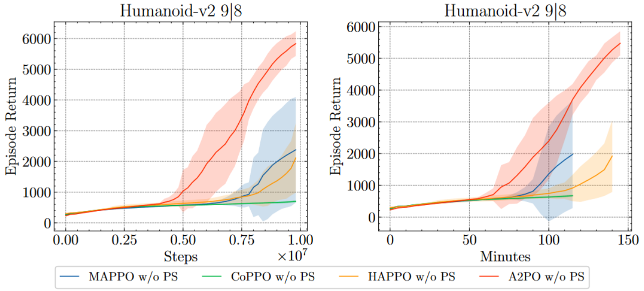
Experiments

- **Ablation Study**

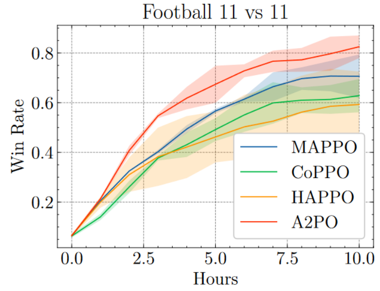


Experiments

- **Training Duration**



(a) Comparison on Humanoid 9|8 over both environment steps and training time



(b) Comparison on GRF 11-vs-11 scenario

Table 6: The comparison of training duration. The format of the first line in a cell is: Training time(Sampling time+Updating Time). The second line of a cell represents the time normalized.

Task	MAPPO	CoPPO	HAPPO	A2PO
3s5z	3h29m(3h3m+0h26m) 1.00(0.87 + 0.13)	3h33m(3h6m+0h27m) 1.02(0.89 + 0.13)	3h49m(3h7m+0h42m) 1.10(0.89 + 0.20)	4h32m(3h41m+0h51m) 1.30(1.06 + 0.25)
27m vs 30m	13h23m(8h31m + 4h52m) 1.00(0.64 + 0.36)	13h19m(8h24m + 4h55m) 1.00(0.63 + 0.37)	16h2m(8h20m + 7h42m) 1.20(0.62 + 0.58)	15h53m(8h7m + 7h46m) 1.19(0.61 + 0.58)
Humanoid 9|8	2h0m(1h45m + 0h15m) 1.00(0.87 + 0.13)	1h58m(1h43m + 0h15m) 0.99(0.86 + 0.13)	2h15m(1h45m + 0h30m) 1.12(0.87 + 0.25)	2h31m(2h0m + 0h31m) 1.26(1.00 + 0.26)
Ant 4x2	6h42m(6h16m + 0h26m) 1.00(0.93 + 0.07)	6h45m(6h19m + 0h26m) 1.01(0.94 + 0.07)	7h29m(6h5m + 1h24m) 1.12(0.91 + 0.21)	7h2m(5h34m + 1h28m) 1.05(0.83 + 0.22)
Humanoid 17x1	12h9m(10h6m + 2h3m) 1.00(0.83 + 0.17)	17h7m(15h5m + 2h2m) 1.41(1.24 + 0.17)	16h55m(11h2m + 5h53m) 1.39(0.91 + 0.48)	19h25m(11h59m + 7h26m) 1.60(0.99 + 0.61)
Football 5vs5	34h46m(32h47m + 1h59m) 1.00(0.94 + 0.06)	32h46m(30h49m + 1h57m) 0.94(0.89 + 0.06)	39h26m(31h54m + 7h32m) 1.13(0.92 + 0.22)	37h26m(30h2m + 7h24m) 1.08(0.86 + 0.21)