The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games

Presented by :
Atmani Hanan

University Mohammed VI Polytechnic

September 4, 2024

Presentation plan

Introduction and motivation

- Proximal Policy Optimization (PPO) has become a leading algorithm in reinforcement learning due to its balance of simplicity and effectiveness.
- Multi-agent environments present unique challenges such as coordination and stability, which are critical for effective learning.
- While PPO is successful in single-agent settings, its performance and effectiveness in multi-agent scenarios need thorough evaluation.
- We aim to assess the performance of PPO and its multi-agent variants IPPO and MAPPO, and identify best practices for applying these algorithms to cooperative multi-agent tasks.

Background

- We study decentralized partially observable Markov decision processes (DEC-POMDP) with shared rewards. A DEC-POMDP is defined by $\langle \mathcal{S}, \mathcal{A}, O, R, P, n, \gamma \rangle$.
- $\mathcal{S}$ is the state space, and $\mathcal{A}$ is the shared action space for each agent i.
- $o_i = O(s; i)$ is the local observation for agent i at global state s.
- $P(s' \mid s, A)$ denotes the transition probability from s to s' given the joint action $A = (a_1, \ldots, a_n)$ for all n agents.
- $R(s, A)$ denotes the shared reward function, and γ is the discount factor. Agents use a policy $\pi_\theta(a_i \mid o_i)$ to produce an action a_i and jointly optimize the discounted accumulated reward $J(\theta) = \mathbb{E}_{A^t, s^t}\left[\sum_t \gamma^t R(s^t, A^t)\right]$ where $A^t = (a_1^t, \ldots, a_n^t)$ is the joint action at time step t.

PPO

- **Independent Proximal Policy Optimization** : Uses PPO to train local policies π_θ and value functions $V_\phi(s)$ for each agent independently, without access to global information.
- **Multi-Agent Proximal Policy Optimization** : Uses PPO with a centralized policy and value function, where the value function can incorporate global information to optimize the performance of all agents collectively.
- **Value Clipping:** In addition to clipping the policy updates, our methods (IPPO and MAPPO) also uses value clipping to restrict the update of critic function for each agent a to be smaller than ϵ using:

$$\mathcal{L}(\phi) = \mathbb{E}_{s_t}\left[\min\left\{\left(V_\phi\left(s_t\right) - \hat{V}_t\right)^2,\right.\right.$$
$$\left.\left.\left(V_{\phi_{old}}\left(s_t\right) + \text{clip}\left(V_\phi\left(s_t\right) - V_{\phi_{old}}\left(s_t\right), -\epsilon, +\epsilon\right) - \hat{V}_t\right)^2\right.$$

- Where ϕ_{old} are old parameters before the update

Multi-agent Particle-world Environment (MPE) Testbed

- **Experimental Setting:** We evaluate three cooperative tasks: Spread, Reference, and Comm. For MAPPO and off-policy methods, a global state is created by combining the agents' local observations. Parameter sharing is not used for Comm due to the diversity of the agents.

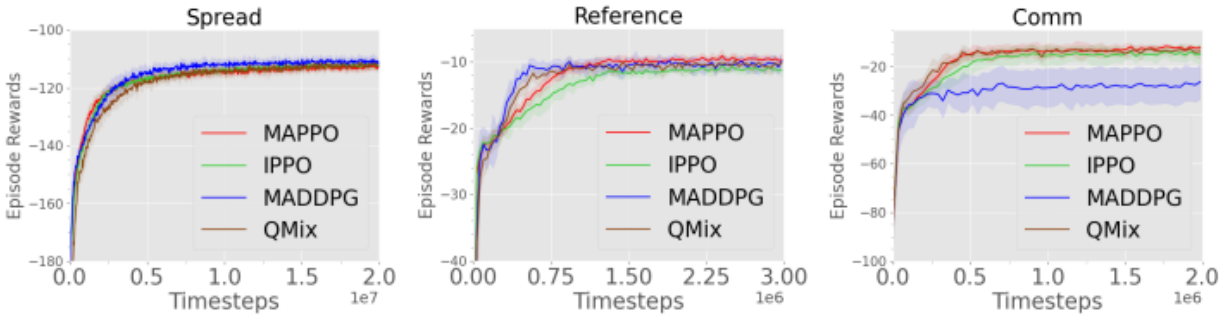

Figure: Performance of different algorithms in the MPEs

- MAPPO and IPPO both show comparable or superior performance to off-policy methods, with MAPPO outperforming IPPO in some tasks.

StarCraft Multi-Agent Challenge (SMAC) Testbed

- **Experimental Setting:**We evaluate three cooperative tasks (Spread, Reference, Comm) using a global state formed by concatenating agents' local observations.

Map	MAPPO(FP)	MAPPO(AS)	IPPO	QMix	RODE*	MAPPO*(FP)	MAPPO*(AS)
2m vs_1z	**100.0**(0.0)	**100.0**(0.0)	**100.0**(0.0)	**95.3**(5.2)	/	*100.0*(0.0)	*100.0*(0.0)
3m	**100.0**(0.0)	**100.0**(1.5)	**100.0**(0.0)	96.9(1.3)	/	*100.0*(0.0)	*100.0*(1.5)
2svs1sc	**100.0**(0.0)	**100.0**(0.0)	**100.0**(1.5)	96.9(2.9)	*100.0*(0.0)	*100.0*(0.0)	*100.0*(0.0)
2s3z	**100.0**(0.7)	**100.0**(1.5)	**100.0**(0.0)	95.3(2.5)	*100.0*(0.0)	96.9(1.5)	96.9(1.5)
3svs3z	**100.0**(0.0)	**100.0**(0.0)	**100.0**(0.0)	**96.9**(12.5)	/	*100.0*(0.0)	*100.0*(0.0)
3svs4z	**100.0**(1.3)	**98.4**(1.6)	**99.2**(1.5)	**97.7**(1.7)	/	*100.0*(2.1)	*100.0*(1.5)
so many baneling	**100.0**(0.0)	**100.0**(0.7)	**100.0**(1.5)	96.9(2.3)	/	*100.0*(1.5)	96.9(1.5)
8m	**100.0**(0.0)	**100.0**(0.0)	**100.0**(0.7)	97.7(1.9)	/	*100.0*(0.0)	*100.0*(0.0)
MMM	**96.9**(0.6)	93.8(1.5)	**96.9**(0.0)	**95.3**(2.5)	/	*93.8*(2.6)	*96.9*(1.5)
1c3s5z	**100.0**(0.0)	96.9(2.6)	**100.0**(0.0)	96.1(1.7)	*100.0*(0.0)	*100.0*(0.0)	96.9(2.6)
bane vs bane	**100.0**(0.0)	**100.0**(0.0)	**100.0**(0.0)	**100.0**(0.0)	*100.0*(46.4)	*100.0*(0.0)	*100.0*(0.0)
3svs5z	**100.0**(0.6)	**99.2**(1.4)	**100.0**(0.0)	**98.4**(2.4)	78.9(4.2)	*98.4*(5.5)	*100.0*(1.2)
2cvs64zg	**100.0**(0.0)	**100.0**(0.0)	98.4(1.3)	92.2(4.0)	*100.0*(0.0)	*96.9*(3.1)	95.3(3.5)
8mvs9m	**96.9**(0.6)	**96.9**(0.6)	**96.9**(0.7)	92.2(2.0)	/	*84.4*(5.1)	*87.5*(2.1)
25m	**100.0**(1.5)	**100.0**(4.0)	**100.0**(0.0)	85.9(7.1)	/	*96.9*(3.1)	*93.8*(2.9)
5mvs6m	**89.1**(2.5)	**88.3**(1.2)	**87.5**(2.3)	75.8(3.7)	*71.1*(9.2)	*65.6*(14.1)	*68.8*(8.2)
3s5z	**96.9**(0.7)	**96.9**(1.9)	**96.9**(1.5)	88.3(2.9)	*93.8*(2.0)	71.9(11.8)	53.1(15.4)
10mvs11m	**96.9**(4.8)	**96.9**(1.2)	**93.0**(7.4)	**95.3**(1.0)	*95.3*(2.2)	81.2(8.3)	*89.1*(5.5)
MMM2	**90.6**(2.8)	**87.5**(5.1)	**86.7**(7.3)	**87.5**(2.6)	*89.8*(6.7)	51.6(21.9)	28.1(29.6)
3s5zvs3s6z	**84.4**(34.0)	63.3(19.2)	**82.8**(19.1)	**82.8**(5.3)	*96.8*(25.11)	*75.0*(36.3)	18.8(37.4)
27mvs30m	**93.8**(2.4)	85.9(3.8)	69.5(11.8)	39.1(9.8)	*96.8*(1.5)	*93.8*(3.8)	*89.1*(6.5)
6hvs8z	**88.3**(3.7)	**85.9**(30.9)	**84.4**(33.3)	9.4(2.0)	*78.1*(37.0)	*78.1*(5.6)	*81.2*(31.8)
corridor	**100.0**(1.2)	**98.4**(0.8)	**98.4**(3.1)	84.4(2.5)	*65.6*(32.1)	*93.8*(3.5)	*93.8*(2.8)

Figure: Median evaluation win rate and standard deviation on all the SMAC maps for different methods

- We observe that IPPO and MAPPO with both the AS and FP inputs achieve strong performance in the vast majority of SMAC

Google Research Football (GRF) Testbed

- **Experimental Setting:** MAPPO was evaluated in several GRF academy scenarios, where a team of agents tries to score against scripted opponents. The results are labeled as "MAPPO" in the tables, even though the agents' local observations make MAPPO and IPPO equivalent. The agents share a single reward, which is the sum of individual rewards. The success rate is measured over 100 game rollouts, and the average success rate from the last 10 evaluations is calculated across 6 seeds.

Scen.	MAPPO	QMix	CDS	TiKick
3v.1	**88.03**(1.06)	8.12(2.83)	76.60(3.27)	76.88(3.15)
CA(easy)	**87.76**(1.34)	15.98(2.85)	63.28(4.89)	/
CA(hard)	**77.38**(4.81)	3.22(1.60)	58.35(5.56)	73.09(2.08)
Corner	**65.53**(2.19)	16.10(3.00)	3.80(0.54)	33.00(3.01)
PS	**94.92**(0.68)	8.05(3.66)	**94.15**(2.54)	/
RPS	**76.83**(1.81)	8.08(4.71)	62.38(4.56)	79.12(2.06)

Figure: Average evaluation success rate and standard deviation (over six seeds) on GRF scenarios for different methods.

Hanabi Testbed

- **Experimental Setting:**We evaluate MAPPO and IPPO in the full-scale Hanabi game with varying numbers of players (2-5 players). We compare MAPPO and IPPO to strong off-policy methods, namely Value Decomposition Networks (VDN) and Simplified Action Decoder (SAD), a Q-learning variant that has been successful in Hanabi.

# Players	Metric	MAPPO	IPPO	SAD	VDN
2	Avg.	23.89(0.02)	**24.00**(0.02)	23.87(0.03)	23.83(0.03)
	Best	**24.23**(0.01)	24.19(0.02)	24.01(0.01)	23.96(0.01)
3	Avg.	**23.77**(0.20)	23.25(0.33)	23.69(0.05)	23.71(0.06)
	Best	**24.01**(0.01)	23.87(0.03)	23.93(0.01)	23.99(0.01)
4	Avg.	**23.57**(0.13)	22.52(0.37)	23.27(0.26)	23.03(0.15)
	Best	23.71(0.01)	23.06(0.03)	**23.81**(0.01)	23.79(0.00)
5	Avg.	**23.04**(0.10)	20.75(0.56)	22.06(0.23)	21.28(0.12)
	Best	**23.16**(0.01)	22.54(0.02)	23.01(0.01)	21.80(0.01)

Figure: Best and Average evaluation scores of MAPPO, IPPO, SAD, and VDN on Hanabi-Full. Results are reported over at-least 3 seeds.

PPO Clipping

- We study the impact of PPO clipping strengths, controlled by the ϵ hyperparameter, in SMAC . Note that ϵ is the same for both policy and value clipping. We generally find that with small ϵ terms such as 0.05, MAPPO's learning speed is slowed in several maps, including hard maps such as MMM2 and 3s5z vs. 3s6z. However, final performance when using $\epsilon = 0.05$ is consistently high and the performance is more stable, as demonstrated by the smaller standard deviation in the training curves. We also observe that large ϵ terms such as 0.2, 0.3, and 0.5, which allow for larger updates to the policy and value function per gradient step, often result in sub-optimal performance.

PPO Clipping

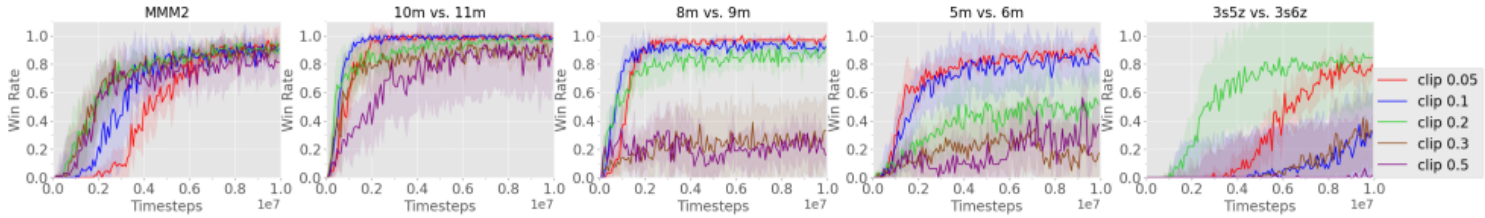

Figure 7: Effect of different clipping strengths on MAPPO's performance in SMAC.

Figure: Effect of different clipping strengths on MAPPO's performance in SMAC

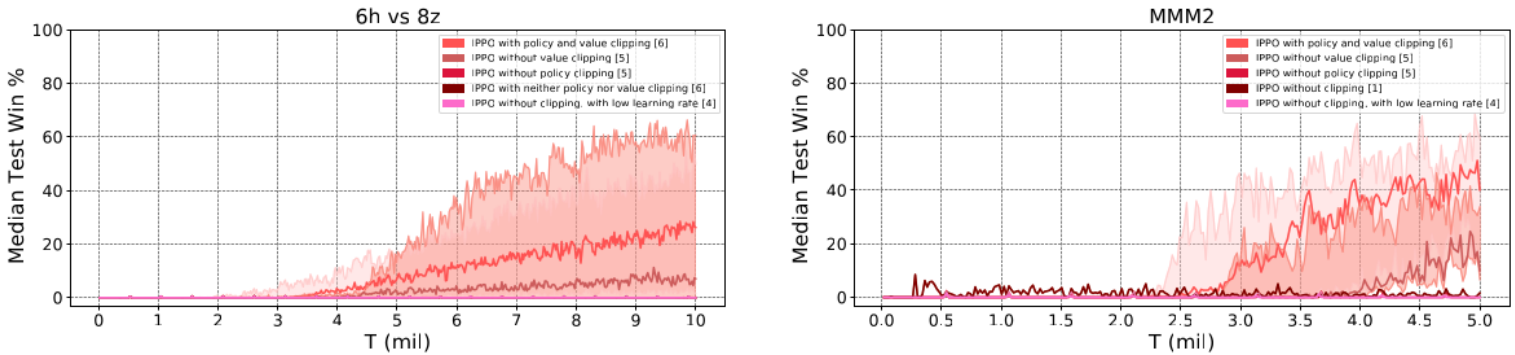

Figure: Ablation study for IPPO with different combinations of policy and value clipping

Conclusion

- PPO and its variants IPPO and MAPPO achieve strong results on cooperative challenges.
- Clipping is crucial for stabilizing learning, with smaller values of ϵ enhancing performance