强化学习原理及应用
Reinforcement Learning (RL): Theories & Applications

DCS6289 Spring 2022

Yucong Zhang （张宇聪）

School of Computer Science and Engineering
Sun Yat-Sen University

Lecture 11: Multi-Agent RL

24th May. 2022

Multi-Agent RL

- Learning cooperation
  - MAPPO
  - EOI

Multi-Agent RL

MAPPO

- On-policy and Off-policy
  - The experiences are collected using the latest learned policy, and then using that experience to improve the policy.
  - Off-policy learning allows the use of older samples(collected using the older policies) in the calculation.
  - On-policy RL algorithms are significantly less sample efficient than off-policy methods.
- Main contributions
  - Multi-Agent PPO(MAPPO), with minimal hyperparameter tuning and without any domain-specific algorithmic changes or architectures, achieves final performances comparable to various off-policy methods
  - MAPPO obtains these strong results while often using a comparable number of samples to many off-policy methods
  - Five implementation and algorithmic factors that govern the practical performance of MAPPO

Yu, Chao, et al. "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games." *arXiv preprint arXiv:2103.01955* (2021).

Multi-Agent RL

MAPPO

- Component
  - Following the CTDE structure: following the algorithmic structure of the single-agent PPO algorithm by learning a policy π_θ and a value function $V_\phi(s)$ which can take extra global information.
  - Generalized Advantage Estimation(GAE)
  - Advantage normalization
  - Observation normalization
  - Gradient clipping
  - Value clipping
  - Layer normalization
  - ReLU activation with orthogonal initialization
  - Large batch size
  - Five concrete implementation details which are insightful and particularly critical to MAPPO's practical performance: value normalization, value function inputs, training data usage, policy and value clipping, death masking.

Yu, Chao, et al. "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games." *arXiv preprint arXiv:2103.01955* (2021).

MAPPO

- Value Normalization
  - To stabilize value learning
  - Standardize the targets of the value function by using running estimates of the average and standard deviation of the value targets
  - Conclusion: using value normalization never hurts training and often significantly improves the final performance of MAPPO

- Training Data Usage
  - A major trick in PPO is the use of importance sampling to perform off-policy corrections, allowing for sample reuse.
  - MAPPO’s performance degrades when samples are re-used too often.
  - Conclusion: avoid using too many training epochs and do not split data into mini-batches.

Yu, Chao, et al. "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games." *arXiv preprint arXiv:2103.01955* (2021).

MAPPO

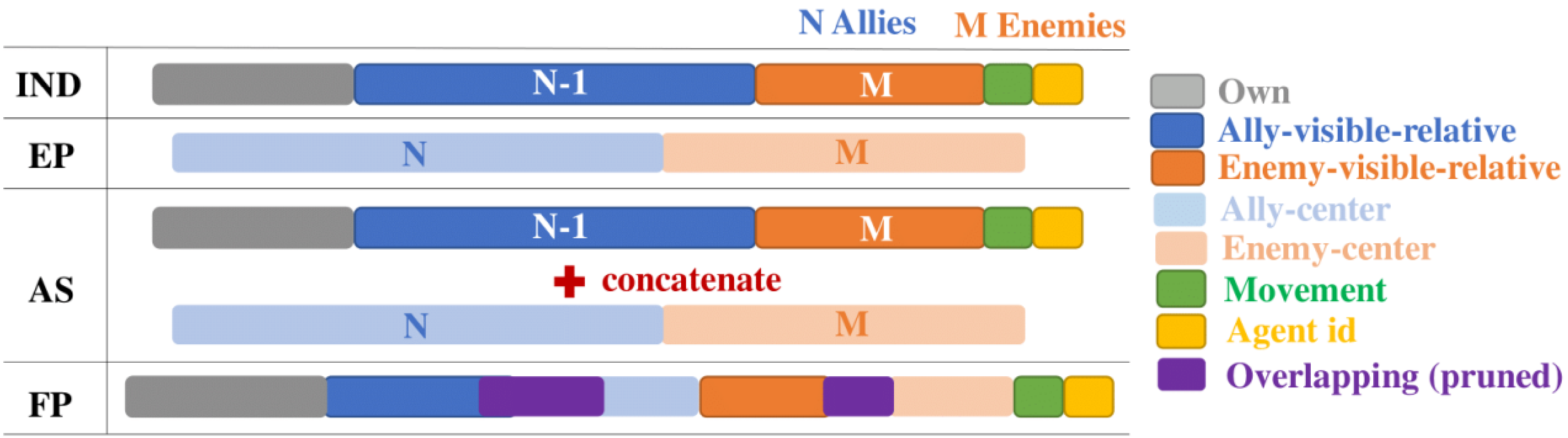
- Input Representation to Value Function
  - Two common implementations for a centralized value function: concatenation of all local observations, an environment-provided global state. Other methods: agent-specific global state representation.
  - Concatenation of all local observations(CL) :
    - the value input dimension can be extremely large when the number of agent or the dimension of local observation is large.
    - CL also may not contain sufficient global information to reduce a POMDP to an MDP.
  - Environment-provided global state(EP): Only contains information common to all agents and can omit important local information
  - Agent-Specific Global State(AS): concatenate the environment state and the local observation
  - Feature-Pruned Agent-Specific Global State(FP): to address the overlap between local and global state, removing all the duplicated features in AS.

Yu, Chao, et al. "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games." *arXiv preprint arXiv:2103.01955* (2021).

Multi-Agent RL

MAPPO

- Input Representation to Value Function
  - IND: using decentralized inputs
  - EP: using environment-provided global state
  - AS: agent-specific global state which concatenates EP and IND
  - FP: removing the duplicated features in AS
  - Conclusion: include agent-specific features in the global state and check that these features do not make the state dimension substantially higher



Yu, Chao, et al. "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games." *arXiv preprint arXiv:2103.01955* (2021).

Multi-Agent RL

MAPPO

- PPO Clipping
  - Clipping the importance ratio and value losses
  - Constrain the policy and value functions from drastically changing between iterations
  - ϵ hyperparameter: lower ϵ values slows learning speed and higher ϵ values result in larger variance and larger volatility in the performance
  - Conclusion: tuning the clipping ratio ϵ as a trade-off between training stability and fast convergence

- Death Masking
  - Using these informative states for dead agents during value learning amplifies the bias of the learned value function
  - Using an agent-specific constant vector, i.e., a zero vector with the agent's ID, as the input to the value function after an agent dies.
  - Conclusion: use zero states with agent ID as the value input for dead agents.

Yu, Chao, et al. "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games." *arXiv preprint arXiv:2103.01955* (2021).

Multi-Agent RL

MAPPO

Algorithm 1 Recurrent-MAPPO

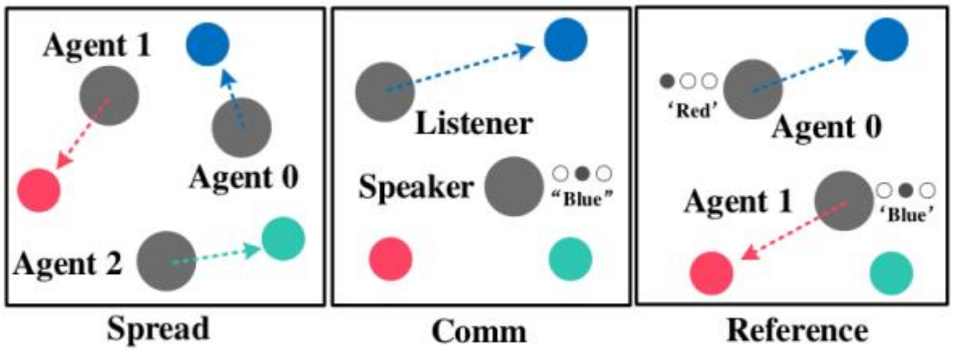
Initialize θ, the parameters for policy π and ϕ, the parameters for critic V, using Orthogonal initialization (Hu et al., 2020)
Set learning rate α
while $step \leq step_{\max}$ **do**
  set data buffer $D = \{\}$
  for $i = 1$ **to** $batch\_size$ **do**
    $\tau = []$ empty list
    initialize $h^{(1)}_{0,\pi}, \ldots h^{(n)}_{0,\pi}$ actor RNN states
    initialize $h^{(1)}_{0,V}, \ldots h^{(n)}_{0,V}$ critic RNN states
    for $t = 1$ **to** T **do**
      for all agents a **do**
        $p^{(a)}_t, h^{(a)}_{t,\pi} = \pi(o^{(a)}_t, h^{(a)}_{t-1,\pi}; \theta)$
        $u^{(a)}_t \sim p^{(a)}_t$
        $v^{(a)}_t, h^{(a)}_{t,V} = V(s^{(a)}_t, h^{(a)}_{t-1,V}; \phi)$
      end for
      Execute actions $\boldsymbol{u_t}$, observe $r_t, s_{t+1}, \boldsymbol{o_{t+1}}$
      $\tau \mathrel{+}= [s_t, \boldsymbol{o_t}, \boldsymbol{h_{t,\pi}}, \boldsymbol{h_{t,V}}, \boldsymbol{u_t}, r_t, s_{t+1}, \boldsymbol{o_{t+1}}]$
    end for
    Compute advantage estimate $\hat{A}$ via GAE on τ, using PopArt
    Compute reward-to-go $\hat{R}$ on τ and normalize with PopArt
    Split trajectory τ into chunks of length L
    for l = 0, 1, .., T//L **do**
      $D = D \cup (\tau[l : l + T, \hat{A}[l : l + L], \hat{R}[l : l + L])$
    end for
  end for
  for mini-batch $k = 1, \ldots, K$ **do**
    $b \leftarrow$ random mini-batch from D with all agent data
    for each data chunk c in the mini-batch b **do**
      update RNN hidden states for π and V from first hidden state in data chunk
    end for
  end for
  Adam update θ on $L(\theta)$ with data b
  Adam update ϕ on $L(\phi)$ with data b
end while

Yu, Chao, et al. "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games." *arXiv preprint arXiv:2103.01955* (2021).

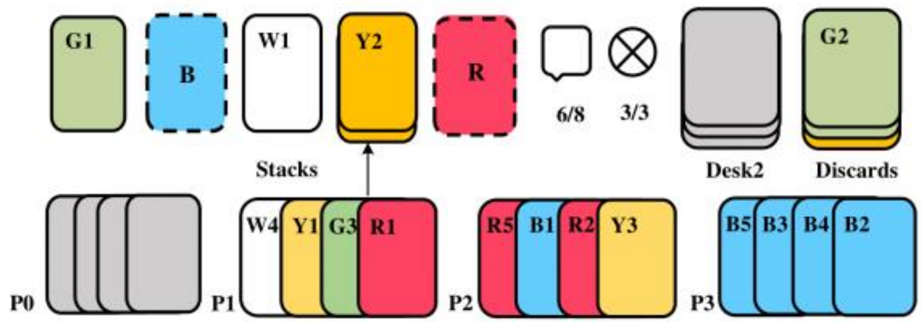
MAPPO

Main Results

- Three multi-agent testbeds: MPE, SC2, Hanabi



(a) MPE scenarios



(b) 4-player Hanabi-Full

(c) SMAC corridor

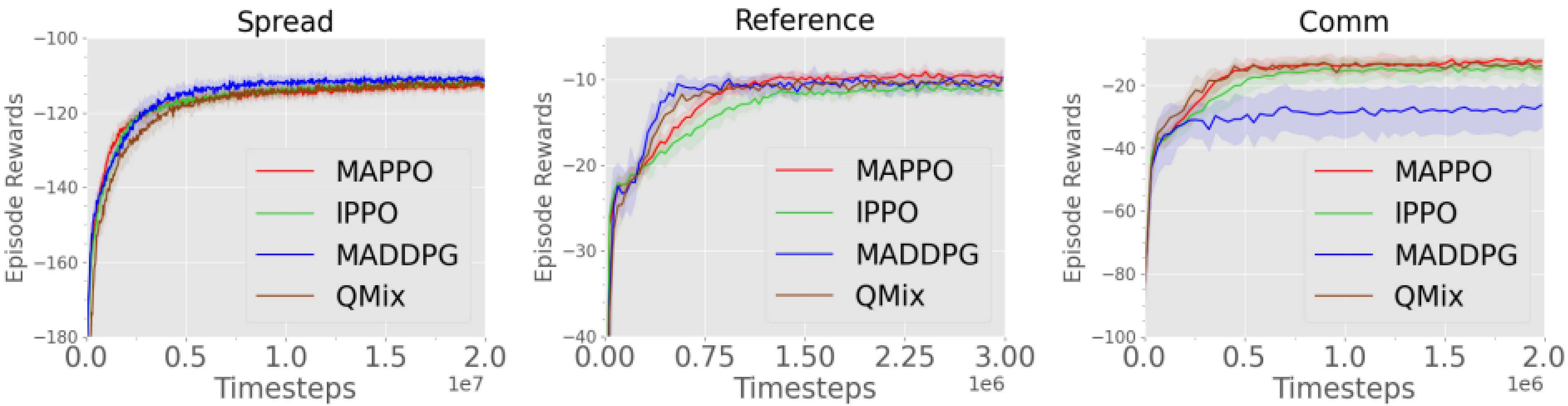
(d) SMAC 2c_vs_64zg

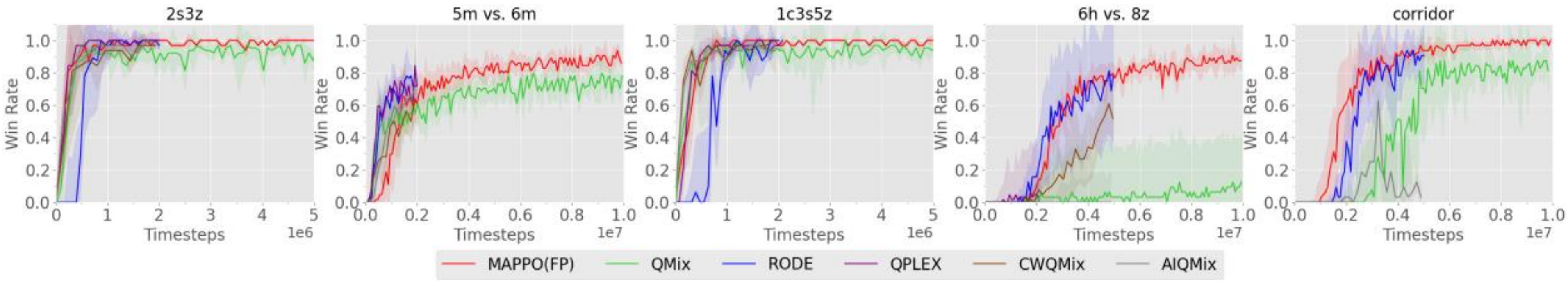
Yu, Chao, et al. "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games." *arXiv preprint arXiv:2103.01955* (2021).

Multi-Agent RL

MAPPO

➢ Main Results





Yu, Chao, et al. "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games." *arXiv preprint arXiv:2103.01955* (2021).

Multi-Agent RL

MAPPO

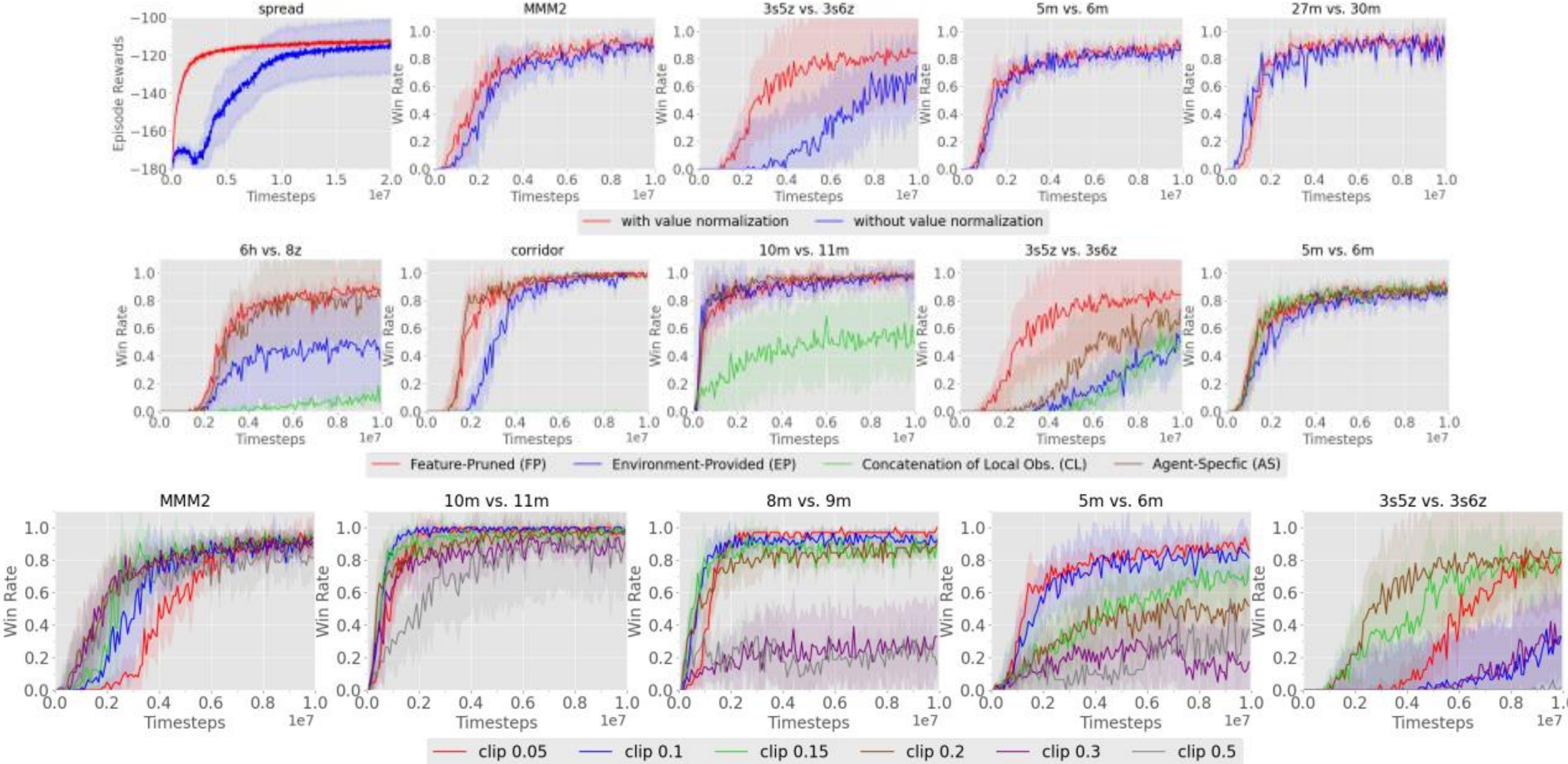
Main Results

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)

Yu, Chao, et al. "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games." *arXiv preprint arXiv:2103.01955* (2021).

MAPPO

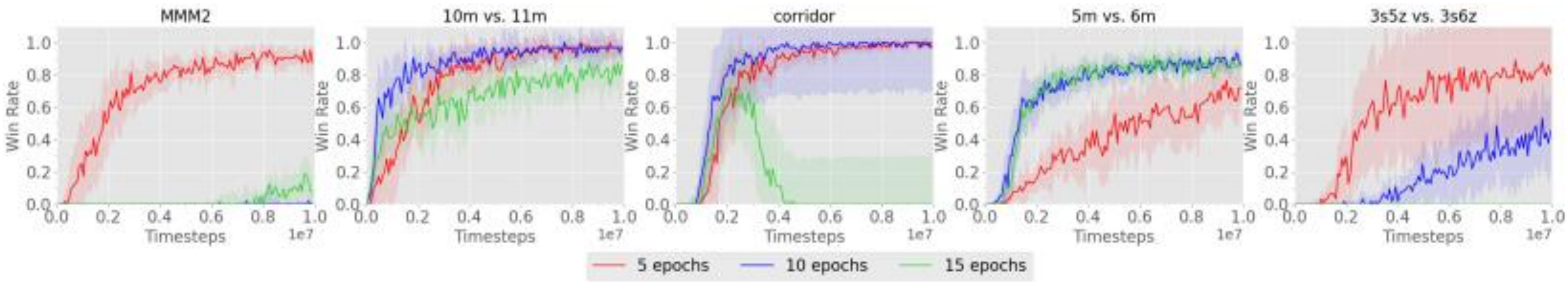
➢ Main Results--Ablation



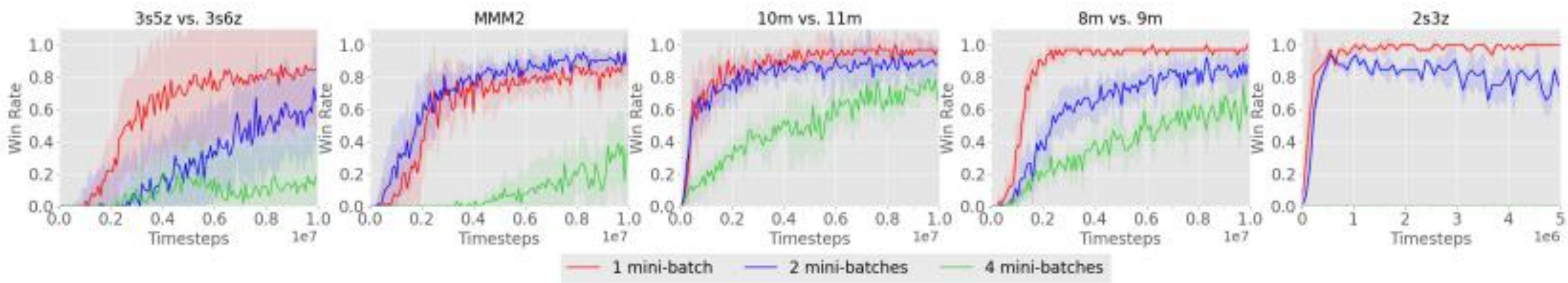
Yu, Chao, et al. "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games." *arXiv preprint arXiv:2103.01955* (2021).

MAPPO

➢ Main Results--Ablation



(a) effect of different training epochs.



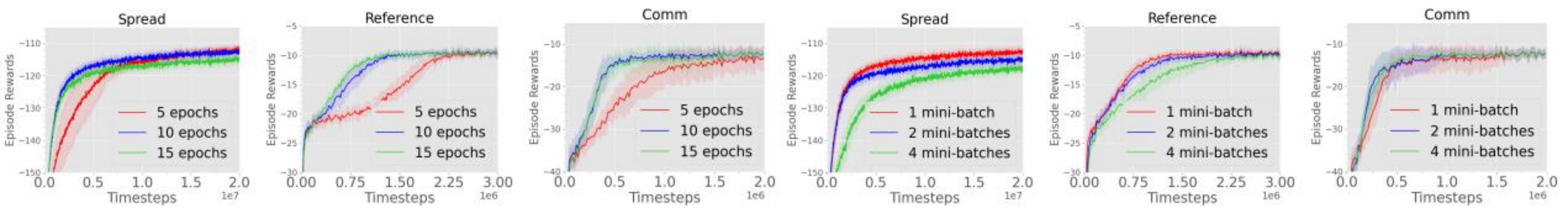
(b) effect of different mini-batch numbers.

Yu, Chao, et al. "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games." *arXiv preprint arXiv:2103.01955* (2021).

Multi-Agent RL

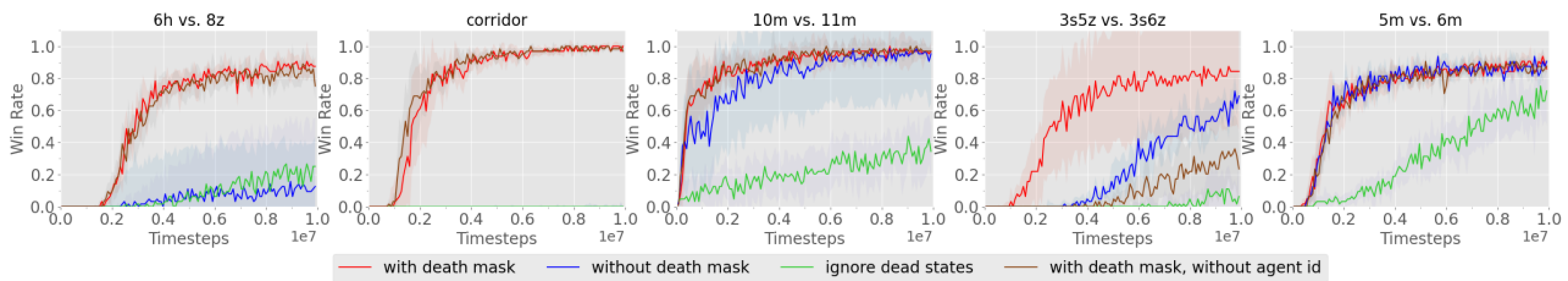
MAPPO

➢ Main Results--Ablation



(a) effect of different training epochs.

(b) effect of different mini-batch numbers.



Yu, Chao, et al. "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games." *arXiv preprint arXiv:2103.01955* (2021).