PPO×Family

第六讲：统筹多智能体

主办 中国计算机学会 CCF

承办 上海人工智能实验室 Shanghai Artificial Intelligence Laboratory

协办 GAIAA 北京大学人工智能研究院 INSTITUTE FOR ARTIFICIAL INTELLIGENCE, PEKING UNIVERSITY 浙江大学 上海高等研究院 SHANGHAI INSTITUTE FOR ADVANCED STUDY ZHEJIANG UNIVERSITY 商汤 sensetime

支持 知乎 机器之心 智海 新一代人工智能科教平台 paperweekly

OpenDILab 浦策

向真实的决策问题前进（多智能体）

生物群体智能

（种群博弈，协同进化）

机器群体智能

（分工协作，各司其职）

Ref: https://www.earth.com/news/fish-swim-in-schools-to-save-energy/
Ref: https://twitter.com/Interior/status/1519073932992778244
Ref: https://www.newscientist.com/article/2357548-us-military-plan-to-create-huge-autonomous-drone-swarms-sparks-concern/
Ref: https://www.nist.gov/programs-projects/performance-collaborative-robot-systems

样例展示

协作任务目标

- 以少胜多，以弱胜强
- 异质智能体之间的配合（不同射程，不同类型）
- 尝试自主发掘到通用的微观操作（集火，走位）

SMAC: https://github.com/oxwhirl/smac
DI-engine + SMAC: https://di-engine-docs.readthedocs.io/zh_CN/latest/13_envs/smac_zh.html

向真实的决策问题前进（多智能体）

常规单智能体决策

恒定的 loss landscape

多智能体决策

动态变化的 loss landscape

MARL Cooperation Overview

多智能体协作基础概述

Ref: https://www.karltuyls.net/wp-content/uploads/2020/06/MA-DM-ICML-ACAI.pdf

多智能体决策的通用设定

多智能体 MDP 的不同点

- 联合动作（joint-action）

  $(a^1, a^2, \cdots, a^N) \in A^1 \times A^2 \times \cdots \times A^N$

- 奖励（agent-wise reward）

  $R^i = R^i(s, a^{-i}, a^i), a^{-i} = (a^1, \cdots, a^N) \setminus a^i$

- 个体最优：最大化单个智能体的累积折扣奖励
- 但团队最优并不是个体最优的叠加，可能出现协作、竞争及更复杂的情况

Ref: https://www.karltuyls.net/wp-content/uploads/2020/06/MA-DM-ICML-ACAI.pdf

Definition

形式化定义：Dec - POMDP

在纯协作（fully-cooperative）的多智能体问题里，各个智能体共享同样的团队奖励

A Dec-POMDP can be described as follows:

$M = \langle I, S, \{A_i\}, P, R, \{\Omega_i\}, O, h \rangle$

- I, the set of agents
- S, the set of states with initial state s_0
- A_i, the set of actions for agent i, with $A = \times_i A_i$ the set of joint actions
- P, the state transition probabilities: $P(s' | s, a)$, the probability of the environment transitioning to state s' given it was in state s and agents took actions a
- R, the global reward function: $R(s, a)$, the immediate reward the system receives for being in state s and agents taking actions a
- Ω_i, the set of observations for agent i, with $\Omega = \times_i \Omega_i$ the set of joint observations
- O, the observation probabilities: $O(o | s, a)$, the probability of agents seeing observations o, given the state is s and agents take actions a
- h, the horizon, whether infinite or if finite, a positive integer
- when h is infinite a discount factor, $0 \leq \gamma < 1$, is used

Ref: http://rbr.cs.umass.edu/camato/decpomdp/overview.html

CTDE：集中式训练分布式执行

以多智能体设定下的 Actor-Critic 算法为例

CTDE Framework

DTDE Framework

Value Decomposition

CTDE 系列方法一：值分解

思路一：将团队的值函数通过某种方式**分解**为个体值函数的组合，再进行求解

即满足 IGM 条件：

(individual-global-max)

$$\boxed{\arg\max_{\boldsymbol{a}\in\mathcal{A}} Q_{tot}(\boldsymbol{s},\boldsymbol{a}) = \left(\arg\max_{a_1\in\mathcal{A}} Q_1(s_1,a_1),\ldots,\arg\max_{a_n\in\mathcal{A}} Q_n(s_n,a_n)\right)}$$

优化结合 TD-learning：

$$\mathscr{L}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a},r,\boldsymbol{s}')\in D}\left[\left(r+\gamma V(\boldsymbol{s}';\boldsymbol{\theta}^-) - Q_{tot}(\boldsymbol{s},\boldsymbol{a};\boldsymbol{\theta})\right)^2\right]$$

- VDN：个人值函数直接相加
- QMIX：保证各分解项的单调性
- Others：更多复杂的设计

VDN: https://arxiv.org/pdf/1706.05296.pdf
QMIX: https://arxiv.org/pdf/1803.11485.pdf
值分解系列方法补充材料：https://github.com/opendilab/PPOxFamily/tree/main/chapter6_marl/chapter6_supp_value_dec.pdf

值分解方法可能失效的情形

特殊的 predator-prey

特殊的 two-state Dec-POMDP

QTRAN: https://arxiv.org/pdf/1905.05408.pdf
QPLEX: https://arxiv.org/pdf/2008.01062.pdf

初步探索：MAPPO

MAPPO

(a) MPE scenarios (b) 4-player Hanabi-Full

(c) SMAC corridor

(d) SMAC 2c_vs_64zg

(e) GRF academy 3 vs 1 with keeper

MAPPO: https://arxiv.org/pdf/2103.01955.pdf

CTDE 系列方法二：MAPG

Independent PG

$$\nabla J_\theta = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{T_n-1} Q_{\phi^i}(s_t, a_t^i)\nabla log p_{\theta^i}(a_t^i|s_t)$$

Multi-Agent PG

$$\nabla J_\theta = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{T_n-1} Q_{\phi^i}(s_t, a_t^{-i}, a_t^i)\nabla log p_{\theta^i}(a_t^i|s_t)$$

MAPG 方法的种种问题

高方差问题（high variance）

Theorem 1. *The CTDE and DT estimators of MAPG satisfy*

随着 n 线性增长

$$\mathbf{Var}_{s_{0:\infty}\sim d_{\boldsymbol{\theta}}^{0:\infty},\mathbf{a}_{0:\infty}\sim\boldsymbol{\pi}_{\boldsymbol{\theta}}}\left[\mathbf{g}_C^i\right]-\mathbf{Var}_{s_{0:\infty}\sim d_{\boldsymbol{\theta}}^{0:\infty},\mathbf{a}_{0:\infty}\sim\boldsymbol{\pi}_{\boldsymbol{\theta}}}\left[\mathbf{g}_D^i\right]\leq\frac{B_i^2}{1-\gamma^2}\sum_{j\neq i}\epsilon_j^2\leq(n-1)\frac{(\epsilon B_i)^2}{1-\gamma^2}$$

where $B_i=\sup_{s,\boldsymbol{a}}\left|\left|\nabla_{\theta^i}\log\pi_{\boldsymbol{\theta}}^i\left(\mathrm{a}^i|\mathrm{s}\right)\right|\right|$, $\epsilon_i=\sup_{s,\boldsymbol{a}^{-i},a^i}\left|A_{\boldsymbol{\theta}}^i(s,\boldsymbol{a}^{-i},a^i)\right|$, *and* $\epsilon=\max_i\epsilon_i$.

baseline 技巧

Theorem 3 (Optimal baseline for MAPG). *The optimal baseline (OB) for the MAPG estimator is*

$$b^{optimal}(\mathrm{s},\mathbf{a}^{-i})=\frac{\mathbb{E}_{\mathrm{a}^i\sim\pi_{\boldsymbol{\theta}}^i}\left[\hat{Q}(\mathrm{s},\mathbf{a}^{-i},\mathrm{a}^i)\left|\left|\nabla_{\theta^i}\log\pi_{\boldsymbol{\theta}}^i(\mathrm{a}^i|\mathrm{s})\right|\right|^2\right]}{\mathbb{E}_{\mathrm{a}^i\sim\pi_{\boldsymbol{\theta}}^i}\left[\left|\left|\nabla_{\theta^i}\log\pi_{\boldsymbol{\theta}}^i(\mathrm{a}^i|\mathrm{s})\right|\right|^2\right]}\tag{7}$$

value 的替换

$$\hat{X}^i(\mathrm{s},\mathbf{a}^{-i},\mathrm{a}^i)=\hat{Q}(\mathrm{s},\mathbf{a}^{-i},\mathrm{a}^i)-b^*(\mathrm{s},\mathbf{a}^{-i}).$$

信用分配问题（credit assignment）

共享的全局信息 Critic 存在信用分配问题，无法给每个智能体准确的优化信号（贡献度）

Settling the Variance of Multi-Agent Policy Gradients: https://arxiv.org/pdf/2108.08612.pdf
QPLEX: https://arxiv.org/pdf/2008.01062.pdf

理论：MAPPO+CTDE 概述

MAPPO：将 PPO 初步拓展到多智能体场景

- 全局（中心化）的 Critic，局部（去中心化）的 Actor
- Actor 部分优化使用标准的 PPO，Critic 部分使用 TD-learning

理论：MAPPO+CTDE●Critic设计

Critic Input 设计：Agent-Specific Global State

- 需要为每个智能体准备不同的 critic 输入信息，目的在于产生智能体专属的 value
- 不同的 global state 编码设计也会带来显著的性能影响

MAPPO: https://arxiv.org/pdf/2103.01955.pdf

代码：一键切换IPPO和MAPPO

原理

- 由于 MAPPO 在优化上基本照搬了 PPO的核心模块，因此整体计算仅需添加并行的 Agent 维度
- 不同的多智能体环境，仅需要定义好全部和局部观测信息 + 相应的 Actor-Critic 神经网络即可

N is the observation shape while M is the action shape

代码样例

```python
from ding.bonus import PPOF


# single-agent RL task
def lunarlander_discrete():
    # Please install lunarlander env first, `pip3 install box2d`
    agent = PPOF(env='lunarlander_discrete', exp_name='./lunarlander_discrete_demo')
    agent.train(step=int(1e5))
    # Batch (Vectorized) evaluation
    agent.batch_evaluate(env_num=4, n_evaluator_episode=8)


# multi-agent RL task
def mpe_simple_spread():
    # Please install pettingzoo[mpe] env first, `pip3 install pettingzoo[mpe]`
    agent = PPOF(env='mpe_simple_spread', exp_name='./mpe_demo', multi_agent=True)
    agent.train(step=int(3e5))
    # Batch (Vectorized) evaluation
    agent.batch_evaluate(env_num=4, n_evaluator_episode=8)
```

完整示例：https://github.com/opendilab/PPOxFamily/tree/main/chapter6_marl/chapter6_application_demo.py
参考文档：https://di-engine-docs.readthedocs.io/zh_CN/latest/04_best_practice/marl_zh.html#di-engine-marl

实践：PPO+MPE

任务目标

- 控制智能体分别覆盖所有的地标，并尽可能避免彼此之间发生碰撞

MDP元素

- 观察：当前智能体的位置和速度+地标和其他智能体的相对位置
- 动作：5维离散动作，空+上下左右
- 奖励：使用所有智能体与地标最小距离的总和+碰撞惩罚作为全局奖励

MPE：https://www.pettingzoo.ml/mpe
DI-engine + MPE：https://di-engine-docs.readthedocs.io/zh_CN/latest/13_envs/pettingzoo_zh.html

实践：PPO+MPE

MAPPO 在 MPE 的各类型环境上，整体都表现得更加强大且稳定

视频样例：https://github.com/opendilab/PPOxFamily/issues/62

多智能体算法总结

- **中心化方法**：学习 $Q^i(s, a^1, a^2, \ldots, a^n)$ 无法避免维数爆炸。
- **值函数分解方法**：值函数分解引理不一定适用所有多智能体场景。
- **策略梯度方法**：策略梯度方差爆炸、多智能体信用分配未知。
- **共享参数方法**：策略探索不足，可随智能体数量变多指数倍变差。

Proposition 1. *Let's consider a fully-cooperative game with an even number of agents n, one state, and the joint action space* $\{0,1\}^n$*, where the reward is given by* $r(\mathbf{0}^{n/2}, \mathbf{1}^{n/2}) = r(\mathbf{1}^{n/2}, \mathbf{0}^{n/2}) = 1$*, and* $r(\boldsymbol{a}^{1:n}) = 0$ *for all other joint actions. Let* $\mathcal{J}^*$ *be the optimal joint reward, and* $\mathcal{J}^*_{share}$ *be the optimal joint reward under the shared policy constraint. Then*

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

理论：回忆 TRPO/PPO 的特点

问题：如何让策略持久稳定提升

解决思路：可靠的方向，合适的步长

$$\eta(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a \mid s) A_\pi(s,a)$$ （新旧策略之间的关系）

$$\eta(\tilde{\pi}) \approx L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a \mid s) A_\pi(s,a)$$ （替代函数）

$$\eta(\tilde{\pi}) \geq L_\pi(\tilde{\pi}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha$$ （新策略和替代函数之间的定量关系）

where $\alpha = \max_s D_{\mathrm{KL}}(\pi(\cdot \mid s) \| \tilde{\pi}(\cdot \mid s)), \epsilon = \max_{s,a} \left| A_\pi(s,a) \right|$

$$\eta(\overline{\pi}) \geq L_\pi(\overline{\pi}) - C\mathrm{D}_{\mathrm{KL}}^{max}(\pi, \overline{\pi})$$
$$\eta(\pi) = L_\pi(\pi) - C\mathrm{D}_{\mathrm{KL}}^{max}(\pi, \pi)$$
$$\rightarrow \eta(\overline{\pi}) \geq \eta(\pi)$$

Proximal policy optimization algorithms
J Schulman, F Wolski, P Dhariwal, A Radford… - arXiv preprint arXiv …, 2017 - arxiv.org
… We propose a new family of **policy gradient** methods for reinforcement learning, which alternate between sampling data through interaction with the environment, and optimizing a "…
☆ Save 99 Cite Cited by 10939 Related articles All 9 versions »

Trust region policy optimization
J Schulman, S Levine, P Abbeel… - … on machine learning, 2015 - proceedings.mlr.press
In this article, we describe a method for optimizing control **policies**, with guaranteed monotonic improvement. By making several approximations to the theoretically-justified scheme, we …
☆ Save 99 Cite Cited by 6163 Related articles All 16 versions »

理论：多智能体优势函数分解引理

Multi-agent state-action value function:

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

- $i_{1:m}$ denotes an ordered subset $\{i_1, \ldots, i_m\}$ of $\mathcal{N}$, and $-i_{1:m}$ refers to its complement.
- i_k refers to the kth agent in the ordered subset.

Multi-agent advantage function:

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

- $j_{1:k}$ and $i_{1:m}$ are disjoint sets.

Lemma 1 (Multi-Agent Advantage Decomposition). *In any cooperative Markov games, given a joint policy $\boldsymbol{\pi}$, for any state s, and any agent subset $i_{1:m}$, the below equations holds.*

$$A_{\boldsymbol{\pi}}^{i_{1:m}}\left(s, \boldsymbol{a}^{i_{1:m}}\right) = \sum_{j=1}^{m} A_{\boldsymbol{\pi}}^{i_j}\left(s, \boldsymbol{a}^{i_{1:j-1}}, a^{i_j}\right).$$

HATRPO/HAPPO: https://arxiv.org/pdf/2109.11251.pdf

追寻本质：优势函数分解引理

The *Multi-Agent Advantage Decomposition Lemma*: $A^{1:3}(s, \boldsymbol{a}^{1:3}) = A^1(s, a^1) + A^2(s, a^1, a^2) + A^3(s, \boldsymbol{a}^{1:2}, a^3)$

Advantage of a^2 *considering* a^1

Advantage of a^3 *considering* $\boldsymbol{a}^{1:2}$

$A^{1,2}(s, \text{🤾}, \text{🥋})$

$= A^1(s, \text{🤾}) + A^2(s, \text{🤾}, \text{🥋})$

π^1 π^2 π^3 : original policy π^1 π^2 π^3 : updated policy

理论：HAPPO 训练流程 ● 伪代码

Algorithm 3 HAPPO

Input: Stepsize α, batch size B, number of: agents n, episodes K, steps per episode T.
Initialize: Actor networks $\{\theta_0^i, \forall i \in \mathcal{N}\}$, Global V-value network $\{\phi_0\}$, Replay buffer $\mathcal{B}$
for $k = 0, 1, \ldots, K-1$ **do**
  Collect a set of trajectories by running the joint policy $\boldsymbol{\pi}_{\boldsymbol{\theta}_0} = (\pi^1_{\theta_0^1}, \ldots, \pi^n_{\theta_0^n})$.
  Push transitions $\{(o_t^i, a_t^i, o_{t+1}^i, r_t), \forall i \in \mathcal{N}, t \in T\}$ into $\mathcal{B}$.
  Sample a random minibatch of B transitions from $\mathcal{B}$.
  Compute advantage function $\hat{A}(\mathrm{s}, \mathbf{a})$ based on global V-value network with GAE.
  Draw a random permutation of agents $i_{1:n}$.
  Set $M^{i_1}(\mathrm{s}, \mathbf{a}) = \hat{A}(\mathrm{s}, \mathbf{a})$.
  for agent $i_m = i_1, \ldots, i_n$ **do**
    Update actor i^m with $\theta_{k+1}^{i_m}$, the argmax of the PPO-Clip objective

$$\frac{1}{BT}\sum_{b=1}^{B}\sum_{t=0}^{T}\min\left(\frac{\pi^{i_m}_{\theta^{i_m}}\left(a_t^{i_m}|o_t^{i_m}\right)}{\pi^{i_m}_{\theta_k^{i_m}}\left(a_t^{i_m}|o_t^{i_m}\right)}M^{i_{1:m}}(s_t, \boldsymbol{a}_t),\ \mathrm{clip}\left(\frac{\pi^{i_m}_{\theta^{i_m}}\left(a_t^{i_m}|o_t^{i_m}\right)}{\pi^{i_m}_{\theta_k^{i_m}}\left(a_t^{i_m}|o_t^{i_m}\right)}, 1 \pm \epsilon\right)M^{i_{1:m}}(s_t, \boldsymbol{a}_t)\right).$$

    Compute $M^{i_{1:m+1}}(\mathrm{s}, \mathbf{a}) = \frac{\pi^{i_m}_{\theta_{k+1}^{i_m}}(\mathrm{a}^{i_m}|\mathrm{o}^{i_m})}{\pi^{i_m}_{\theta_k^{i_m}}(\mathrm{a}^{i_m}|\mathrm{o}^{i_m})}M^{i_{1:m}}(\mathrm{s}, \mathbf{a})$. //Unless $m = n$.
  end for
  Update V-value network by following formula:
  $\phi_{k+1} = \arg\min_\phi \frac{1}{BT}\sum_{b=1}^{B}\sum_{t=0}^{T}\left(V_\phi(s_t) - \hat{R}_t\right)^2$
end for

- **初始化**：计算联合优势函数+选择随机排列
- **优化**：替换原有 PPO 中的优势函数
- **迭代**：结合优势函数分解定理和GAE的计算，迭代更新实际使用的优势函数

HATRPO/HAPPO 补充材料：https://github.com/opendilab/PPOxFamily/tree/main/chapter6_marl/chapter6_supp_happo.pdf

实践：PPO+MA MuJoCo

任务目标

- 测试智能体操控机器人各个部分，组合完成复杂控制任务的能力

MDP元素

- 观察：机器人物理属性构成的向量
- 动作：多维连续动作，范围[-1, 1]
- 奖励：稠密奖励，包含运动速度奖励和机械控制惩罚等部分

MA MuJoCo: https://github.com/schroederdewitt/multiagent_mujoco
DI-engine + MA MuJoCo: https://di-engine-docs.readthedocs.io/zh_CN/latest/13_envs/mamujoco_zh.html

实践：PPO+MA MuJoCo

(a) 2x4-Agent Ant (b) 4x2-Agent Ant (c) 8x1-Agent Ant

(d) 2x3-Agent HalfCheetah (e) 3x2-Agent HalfCheetah (f) 6x1-Agent HalfCheetah

(g) 2x3-Agent Walker (h) 3x2-Agent Walker (i) 6x1-Agent Walker

(j) 17x1-Agent Humanoid (k) 17x1-Agent HumanoidStandup (l) 10x2-Agent Swimmer

Bi-DexHands

Bi-DexHands: https://github.com/PKU-MARL/DexterousHands
视频样例：https://github.com/opendilab/PPOxFamily/issues/62

BAGS OF
OpenDILab
MARL
TRICKS

Trick:MA Transformer ● MAT架构

MAT: https://proceedings.neurips.cc/paper_files/paper/2022/file/69413f87e5a34897cd010ca698097d0a-Paper-Conference.pdf

Trick:MA Transformer ● MAT架构

- The decoder is parameterized by θ;
- Input the embedded joint action $\boldsymbol{a}$, decode, and output a representation of the joint action $\hat{\boldsymbol{a}}$, which is fed to an MLP to produce the policy;
- Minimize the following clipping PPO objective of

$$L_{\text{Decoder}}(\theta) = -\frac{1}{Tn}\sum_{m=1}^{n}\sum_{t=0}^{T-1}\min\left(\mathrm{r}_t^{i_m}(\theta)\hat{A}_t, \operatorname{clip}\left(\mathrm{r}_t^{i_m}(\theta), 1\pm\epsilon\right)\boxed{\hat{A}_t}\right)$$

优势函数分解

$$\text{where } \mathrm{r}_t^{i_m}(\theta) = \frac{\pi_\theta^{i_m}\left(\mathbf{a}_t^{i_m} \mid \hat{\mathbf{o}}_t^{i_{1:n}}, \hat{\mathbf{a}}_t^{i_{1:m-1}}\right)}{\pi_{\theta_{\text{old}}}^{i_m}\left(\mathbf{a}_t^{i_m} \mid \hat{\mathbf{o}}_t^{i_{1:n}}, \hat{\mathbf{a}}_t^{i_{1:m-1}}\right)}$$

MAT: https://proceedings.neurips.cc/paper_files/paper/2022/file/69413f87e5a34897cd010ca698097d0a-Paper-Conference.pdf

Trick: Param. Sharing 共享参数

	# Agents	# Types	Type Distribution
BPS (1)	15	3	5–5–5
BPS (2)	30	3	10–10–10
BPS (3)	30	5	6–6–6–6–6
BPS (4)	30	5	2–2–2–15–9
BPS-h (1)	15	3†	5–5–5
BPS-h (2)	30	5†	6–6–6–6–6
BPS-h (3)	200	4†	50–50–50–50
C-RWARE (1)	4	2‡	2–2
C-RWARE (2)	8	2‡	4–4
C-RWARE (3)	16	2‡	8–8
LBF	12	3	4–4–4–4
MMM2	10§	3	7–2–1

- 同质智能体之间共享参数可以显著提升训练性能
- 异质智能体之间不恰当的共享参数会带来阻碍

Scaling Multi-Agent Reinforcement Learning with Selective Parameter Sharing: https://arxiv.org/pdf/2102.07475.pdf

Trick: Various Mask 掩码

- **存活掩码（death mask）**：当智能体死亡后，将其除 agent id 之外的信息全部置为0
- **动作掩码（action mask）**：给出当前时间步某个智能体可执行的动作集合，即动作空间的子集，从而提升优化效率（课程第二讲作业）

MAPPO: https://arxiv.org/pdf/2103.01955.pdf
A Closer Look at Invalid Action Masking in Policy Gradient Algorithms: https://arxiv.org/pdf/2006.14171.pdf

Trick: Entropy Balance ● 熵

SMAC

3s5z

SMAC

MMM2

- 不合适的 Entropy Bonus 很容易导致策略早熟，策略输出分布的熵在前期急速下降到低值，失去足够的探索能力，从而使得最终获得的智能体性能较差
- 需要根据动作空间的大小相应调整超参数，也可设置自适应衰减的 Entropy Weight

总结：PPO + 多智能体

Summary

小节	算法要点	代码和实践要点
多智能体协作基础概述	• Dec-POMDP 的定义 • CTDE Framework 和 IGM 条件	• IGM 失效的情形
MAPPO	• Multi-Agent PG 的问题 • MAPPO 如何结合 CTDE • MAPPO 如何设计 global state	• 多粒子运动（MPE） • 一键切换 IPPO 和 MAPPO
HAPPO/HATRPO	• 回忆 TRPO/PPO 的特点 • 多智能体优势函数分解引理 • HAPPO 训练流程	• 多智能体机器人控制协作 （Multi-Agent MuJoCo）
Bags of Tricks in MARL	• 使用Transformer • 共享参数 • 动作掩码和存活掩码 • 策略多样性的平衡与控制	• 星际争霸2微观操作（SMAC） • 谷歌足球博弈（GRF）

ACE 补充材料：https://github.com/opendilab/PPOxFamily/tree/main/chapter6_marl/chapter6_supp_ace.pdf

下节预告

（七）PPO 的七重境界：挖掘黑科技

- 权衡偏差与方差：GAE
- 稳扎稳打：Recompute Advantage
- 巧夺天工：Entropy Balance
- 免死金牌：Gradient Norm
- 笨鸟先飞：Network Initialization
- 亡羊补牢：Dual Clip
- 合理运用随机：Dynamic Seed