Diving into Pommerman:
A Multi-agent Playground

DIYA MARL team

Index

1. **Environment**
2. **RL**
   a. Value-based
   b. Policy-based
   c. Results
3. **MARL**
   a. COMA
   b. QMIX
   c. Results
4. **Future work**

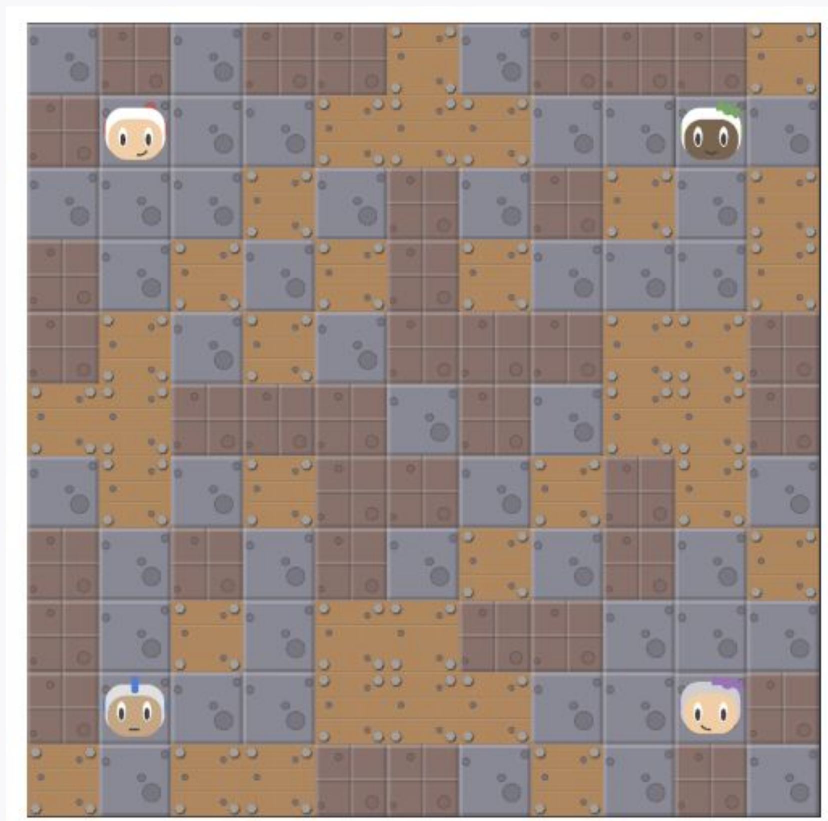
Pommerman

Pommerman: A Multi-Agent Playground

Cinjon Resnick* NYU
Wes Eldridge Rebellious Labs
David Ha Google Brain
Denny Britz Stanford University
Jakob Foerster University of Oxford
Julian Togelius NYU
Kyunghyun Cho and **Joan Bruna** NYU, FAIR

six actions

0: stop
1: up
2: down
3: left
4: right
5: bomb

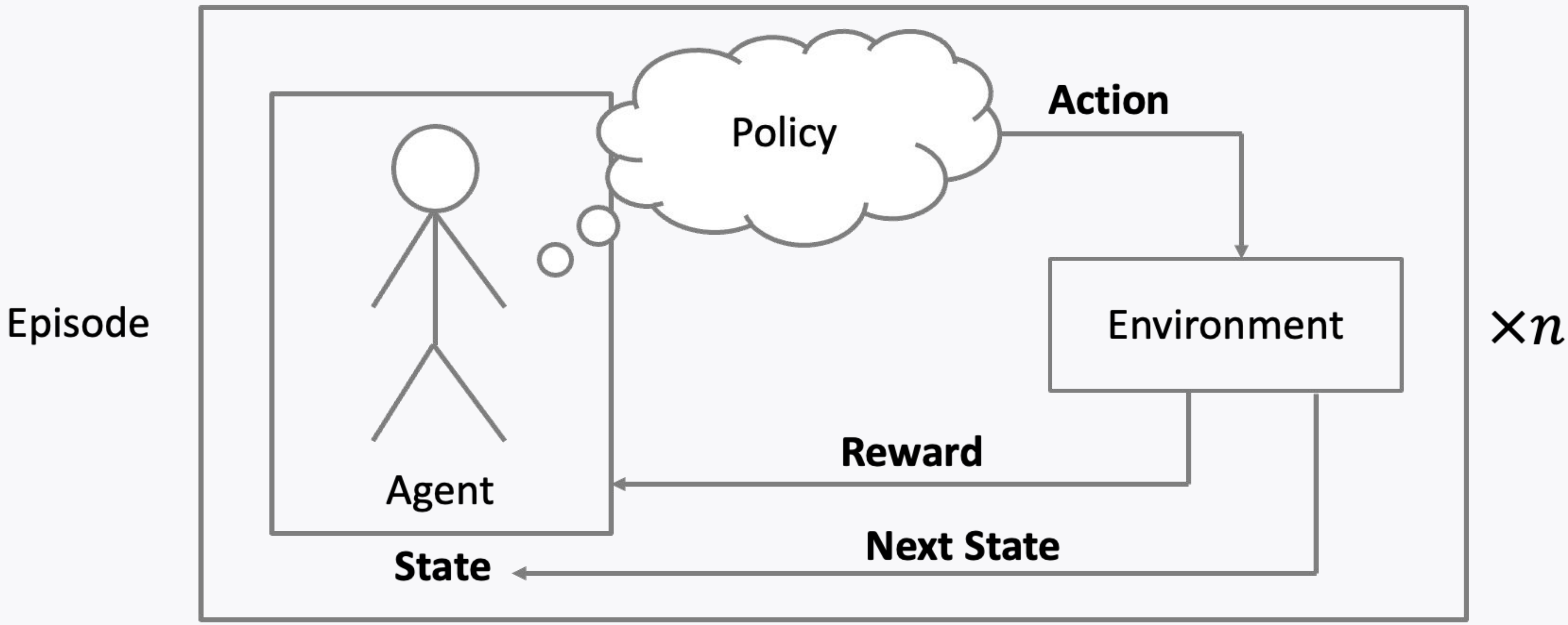


three items:

extra bomb,
can kick,
and extra blast

Reinforcement Learning

Sequential Decision Making through **Trial and Error**



State → Action → Reward, Next State

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-t-1} R_T$$

Value Function

Value Function $$v_\pi(s) = \boldsymbol{E}_\pi[G_t | S_t = s]$$

Bellman Equation $$v_\pi(s) = \boldsymbol{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s]$$

TD Update $$V(S_t) \leftarrow V(S_t) + \alpha \left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right)$$

TD Update for Maximum Return

$$V(S_t) \leftarrow V(S_t) + \alpha \left(\boxed{R_{t+1} + \gamma max V(S_{t+1})} - V(S_t) \right)$$

$$Q(s_t, a_t) = Q(s_t, a_t) + \eta * \left(\boxed{R_{t+1} + \gamma \max_a Q(s_{t+1}, a)} - Q(s_t, a_t) \right)$$

Value-based RL

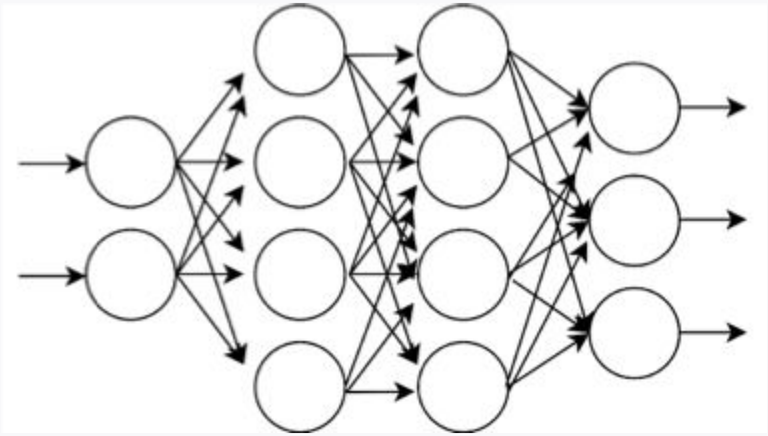
Q function

$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$$

$$Q(s_t, a_t) = Q(s_t, a_t) + \eta * \left(R_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right)$$

Q Network

State s



q(s, a_1)

q(s, a_2)

q(s, a_3)

backprop (Gradient Descent)

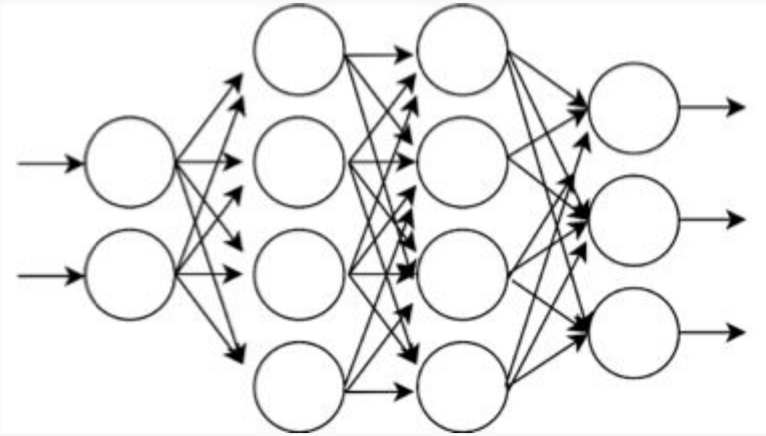
$$E(s_t, a_t) = \left(R_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right)^2$$

Policy-based RL

Policy $\pi(a|s) = \Pr(A_t = a|S_t = s)$

Policy Network

State s



π(a$_1$|s)

π(a$_1$|s)

π(a$_1$|s)

Objective Function: $J(\theta)$

Gradient Ascent

$$\theta' = \theta + \underline{\nabla_\theta J(\theta)}$$

$$= \mathbb{E}[\nabla_\theta \log \pi_\theta \times X]$$

Actor-Critic

Actor

Policy Update $$\theta \leftarrow \theta + \alpha_\theta \underline{Q_w(s,a)\nabla_\theta \ln \pi_\theta(a|s)};$$

Gradient of Policy evaluation

Critic

TD error $$\delta_t = r_t + \gamma Q_w(s',a') - Q_w(s,a)$$

Value Function Update $$w \leftarrow w + \alpha_w \delta_t \nabla_w Q_w(s,a)$$

Soft Actor-Critic

Actor

Policy Update

Maximize Entropy along with Expected Return

$$J(\theta) = \sum_{t=1}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_{\pi_\theta}}[r(s_t, a_t) + \alpha\mathcal{H}(\pi_\theta(.\,|s_t))]$$

Critic

Soft Q-value

Target Q-network

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma\underline{\mathbb{E}_{s_{t+1}\sim\rho_\pi(s)}[V(s_{t+1})]} \quad \text{; according to Bellman equation.}$$

$$\text{where } V(s_t) = \mathbb{E}_{a_t\sim\pi}[Q(s_t, a_t) - \alpha\log\pi(a_t|s_t)] \quad \text{; soft state value function.}$$

objective

$$J_V(\psi) = \mathbb{E}_{s_t\sim D}[\frac{1}{2}\Big(V_\psi(s_t) - \mathbb{E}[Q_w(s_t, a_t) - \log\pi_\theta(a_t|s_t)]\Big)^2]$$

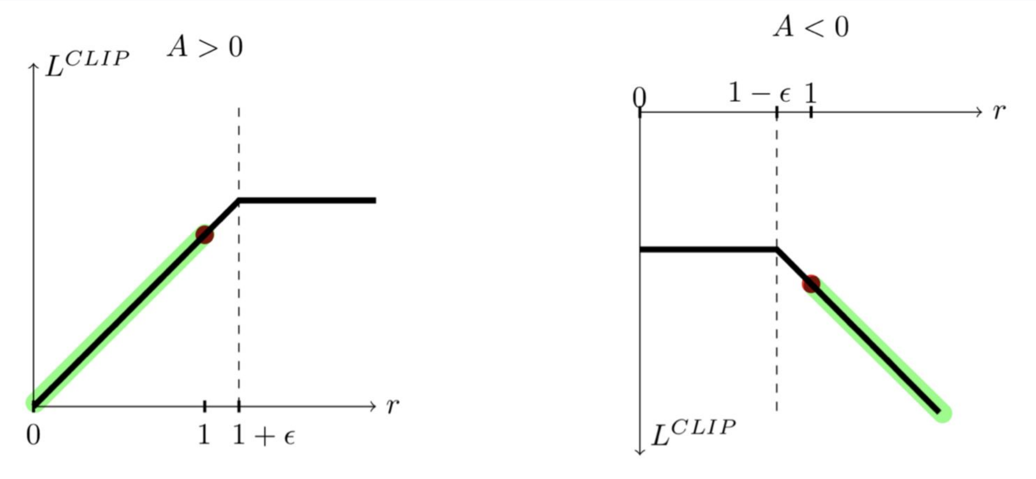
PPO

Policy Ratio

$$r(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}$$

Surrogate objective

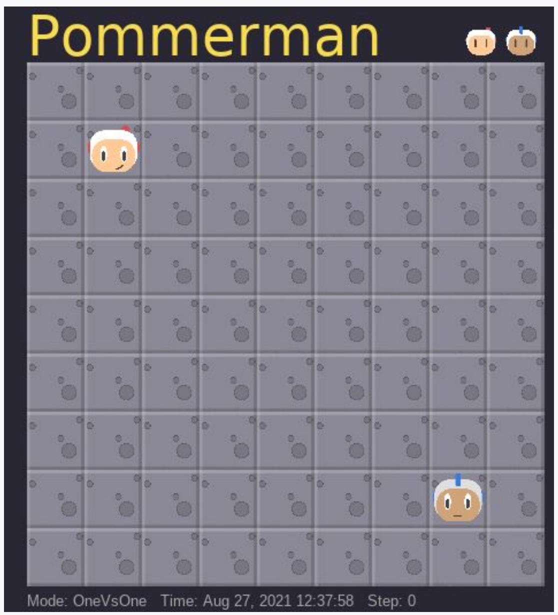
$$J^{\text{CLIP}}(\theta) = \mathbb{E}[\min(r(\theta)\hat{A}_{\theta_{\text{old}}}(s,a), \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{\theta_{\text{old}}}(s,a))]$$

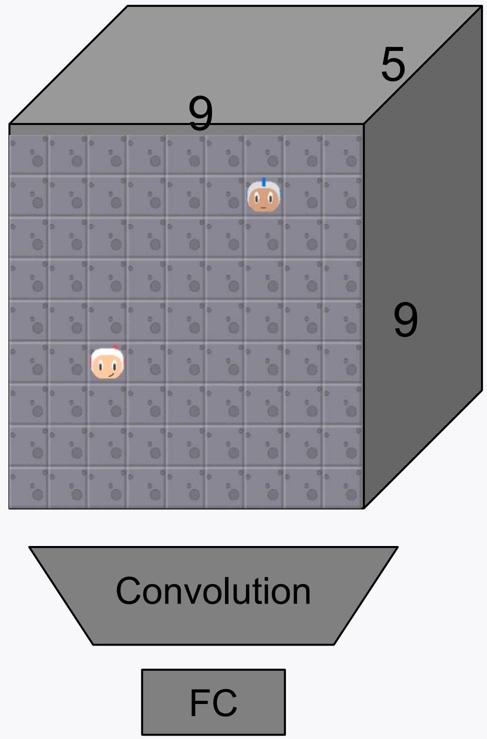


Single play

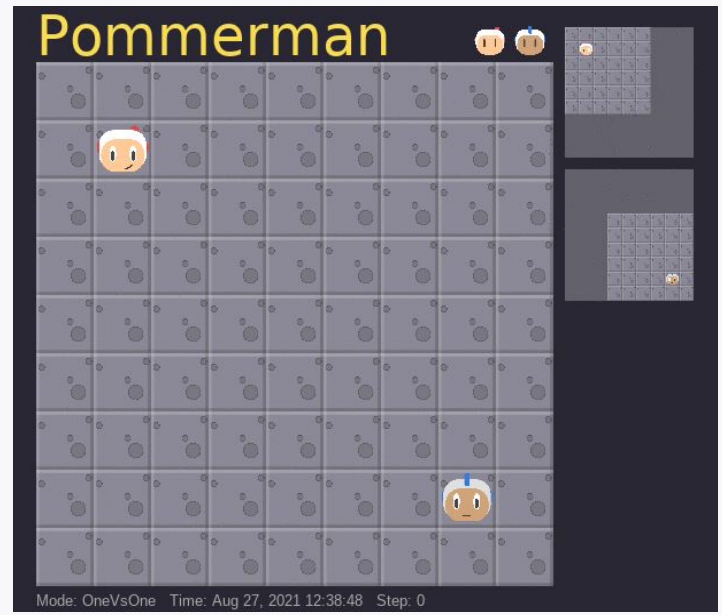
DQN

PPO

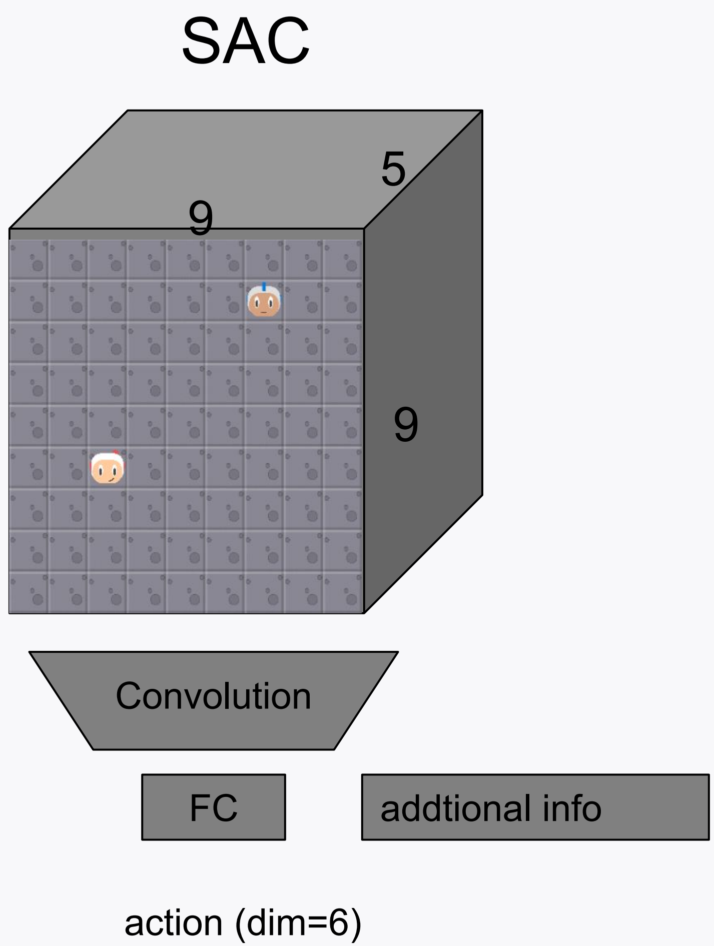




action (dim=6)

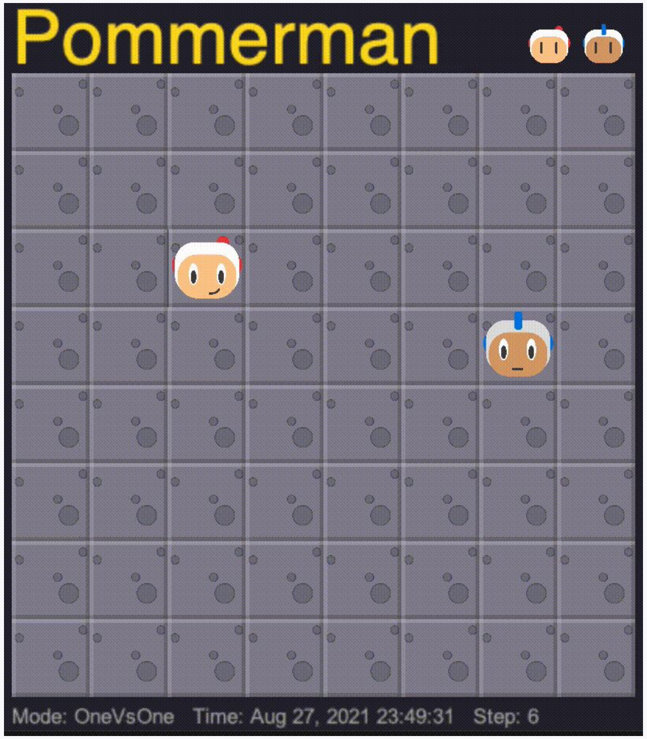


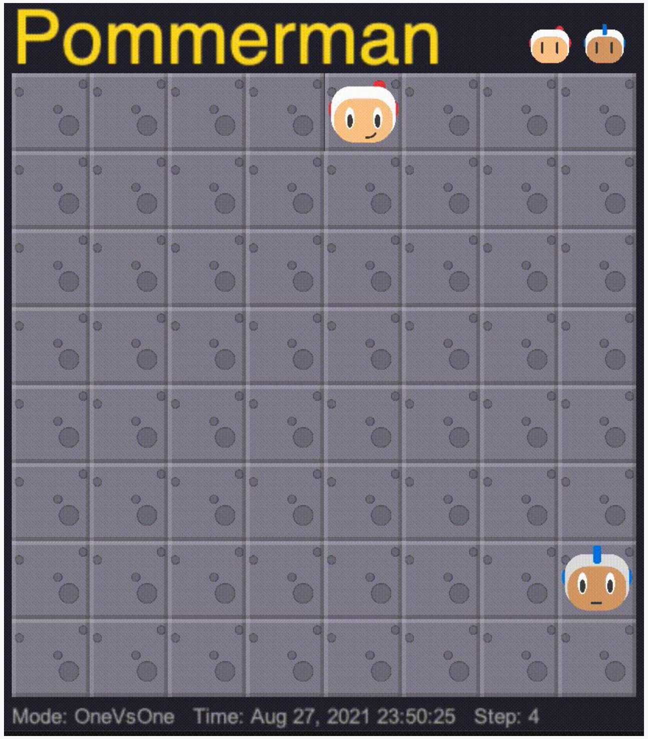
Single play



Bug finding

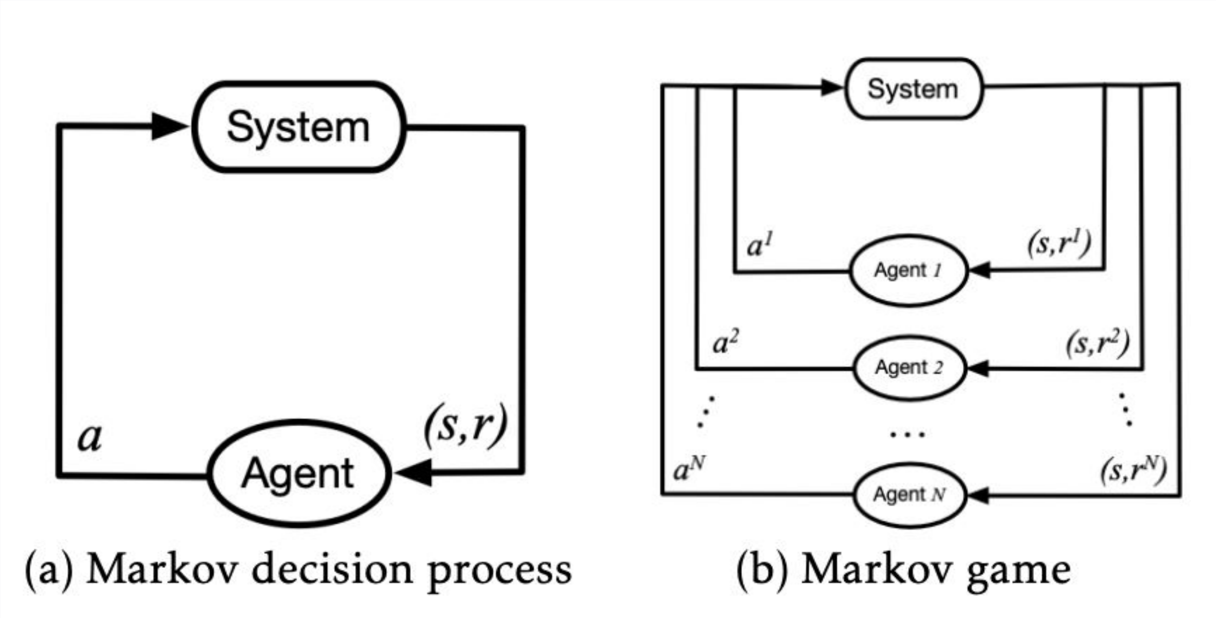
Adversarial attack





MARL

RL vs MARL



(image source: Zhang et. al. 2021)

Joint-action dependency of state transition

Partially Observable Markov Decision Process (POMDP)

- **Competitive / Cooperative Behavior**
- **Learning Curriculum; Appropriate opponent**
- **Emergence Behavior; Self-play → Auto Curricula**

Counterfactual Multi-Agent Policy Gradients

1. Counterfactual

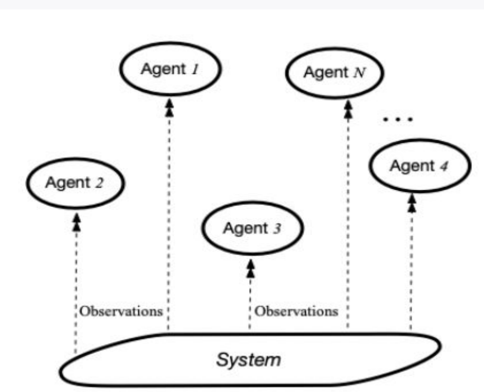
명사

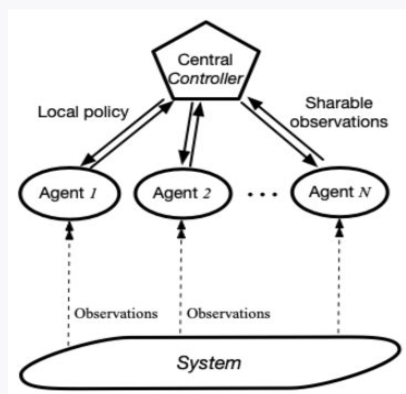
1. 논리
조건법적 서술: 어떤 문장의 첫절이 사실과 정반대인 것을 서술할 경우의 표현법; 예를 들면 「만약 내가 알고 있었다면」(if I had known) 따위.

(image source: https://en.dict.naver.com/#/entry/enko/238ce2832a61426eb59007f28788411e)

Causal Inference 분야에서 자주 등장하는 개념 e.g. 정책평가

2. Independent RL v.s. Centralized Critic & Decentralized Actor





(image source: Zhang et. al. 2021)

COMA concept

Policy gradient

$$g = \mathbb{E}_{s_{0:\infty}, u_{0:\infty}} \left[\sum_{t=0}^{T} \boxed{R_t} \nabla_{\theta^\pi} \log \pi(u_t|s_t) \right]$$

COMA gradient

$$g = \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{a} \nabla_\theta \log \pi^a(u^a|\tau^a) \boxed{A^a(s, \mathbf{u})} \right]$$

나를 제외한 player들의 action

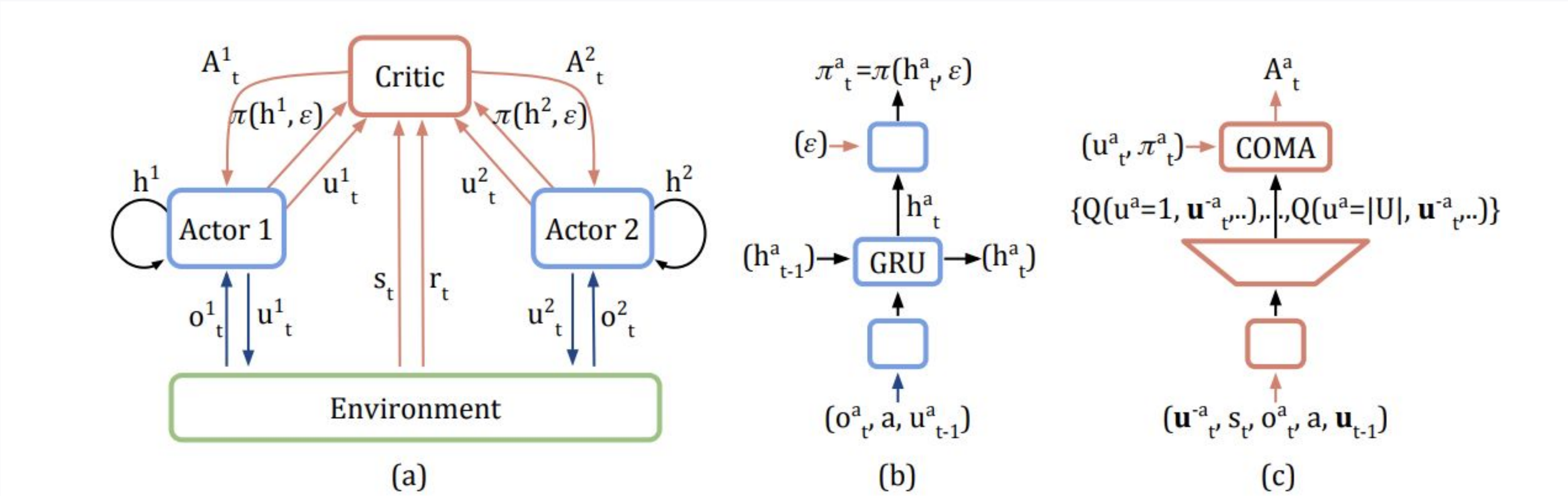
Counterfactual Advantage

$$A^a(s, \mathbf{u}) = Q(s, \mathbf{u}) - \sum_{u'^a} \pi^a(u'^a|\tau^a) Q(s, (\mathbf{u}^{-a}, u'^a)).$$

다른 agent들의 action은 고정된 상태에서 내가 다른 action을 취했더라면…?

이걸 선택했더라면?

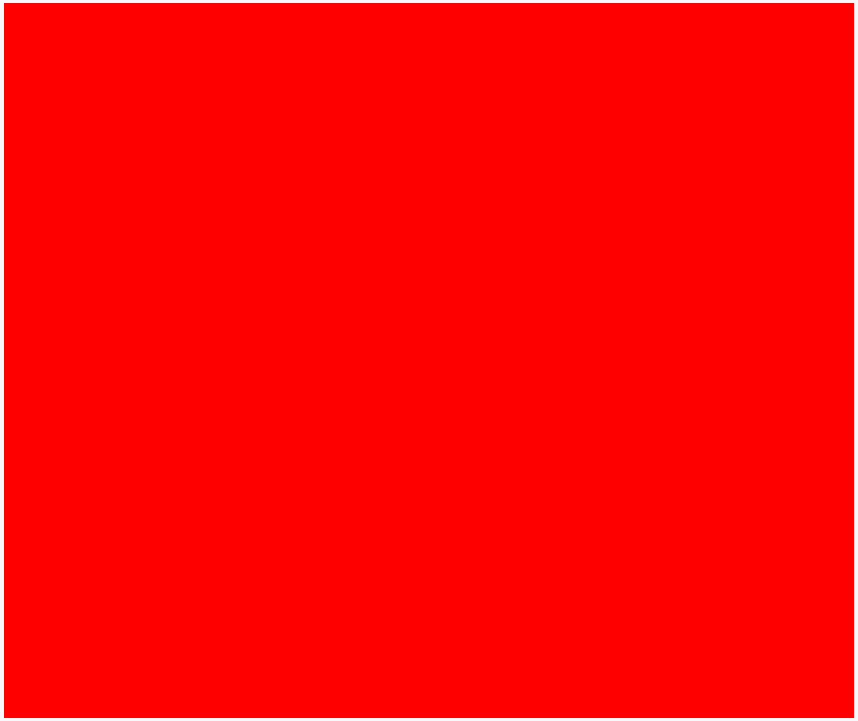
COMA diagram



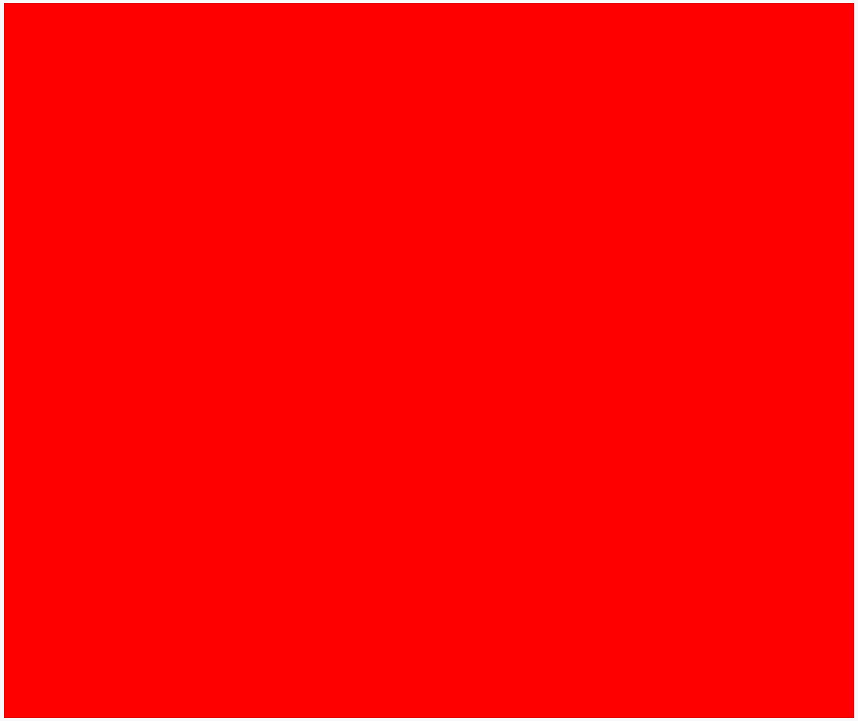
(image source: Foerster et. al. 2018)

COMA result

자폭



평화 협정



QMIX

Cooperative team play

- 팀 단위의 reward
- Decentralized agents

전체 value function은 알겠어
각 agent 별로 Q function을
어떻게 구하지?

내가 몇등인지 보다 팀이 이기는게 중요하다

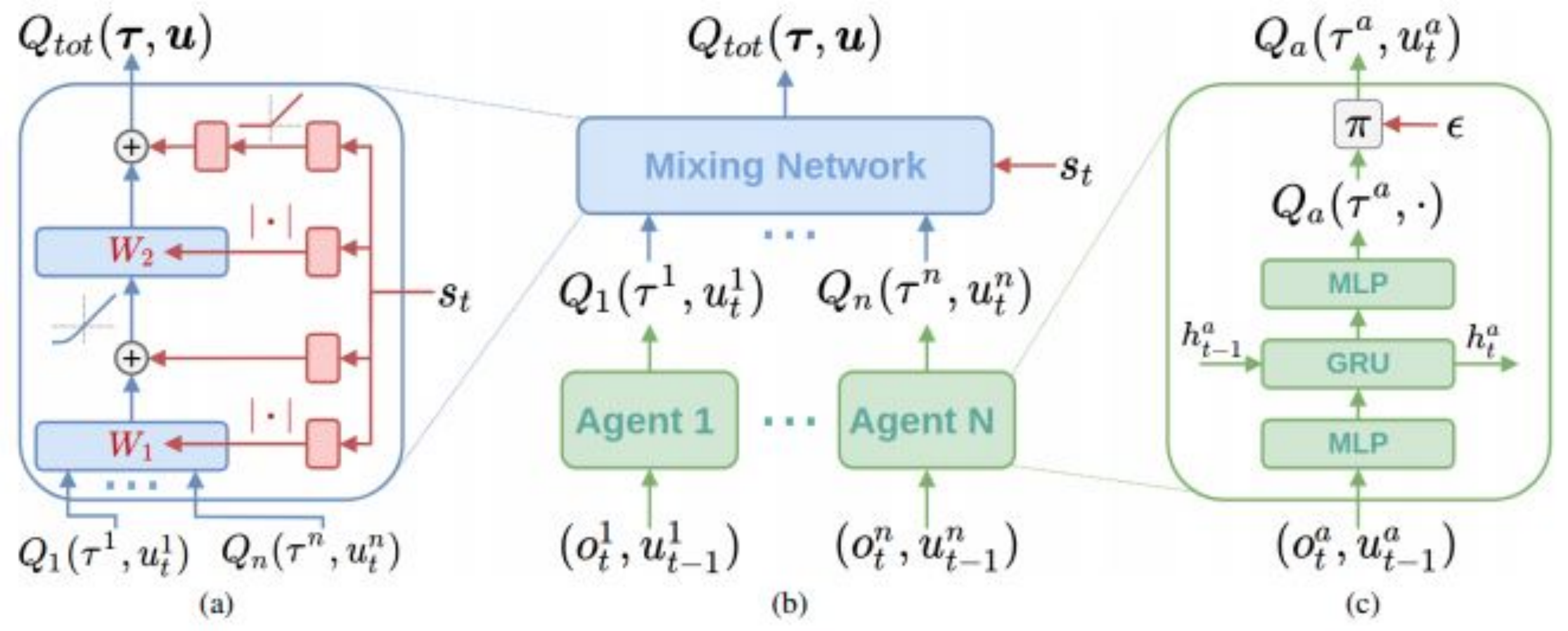
각 플레이어가 팀의 모든 **observation**을 볼 수 없다

QMIX concept

Monotonicity Q value among cooperative agents

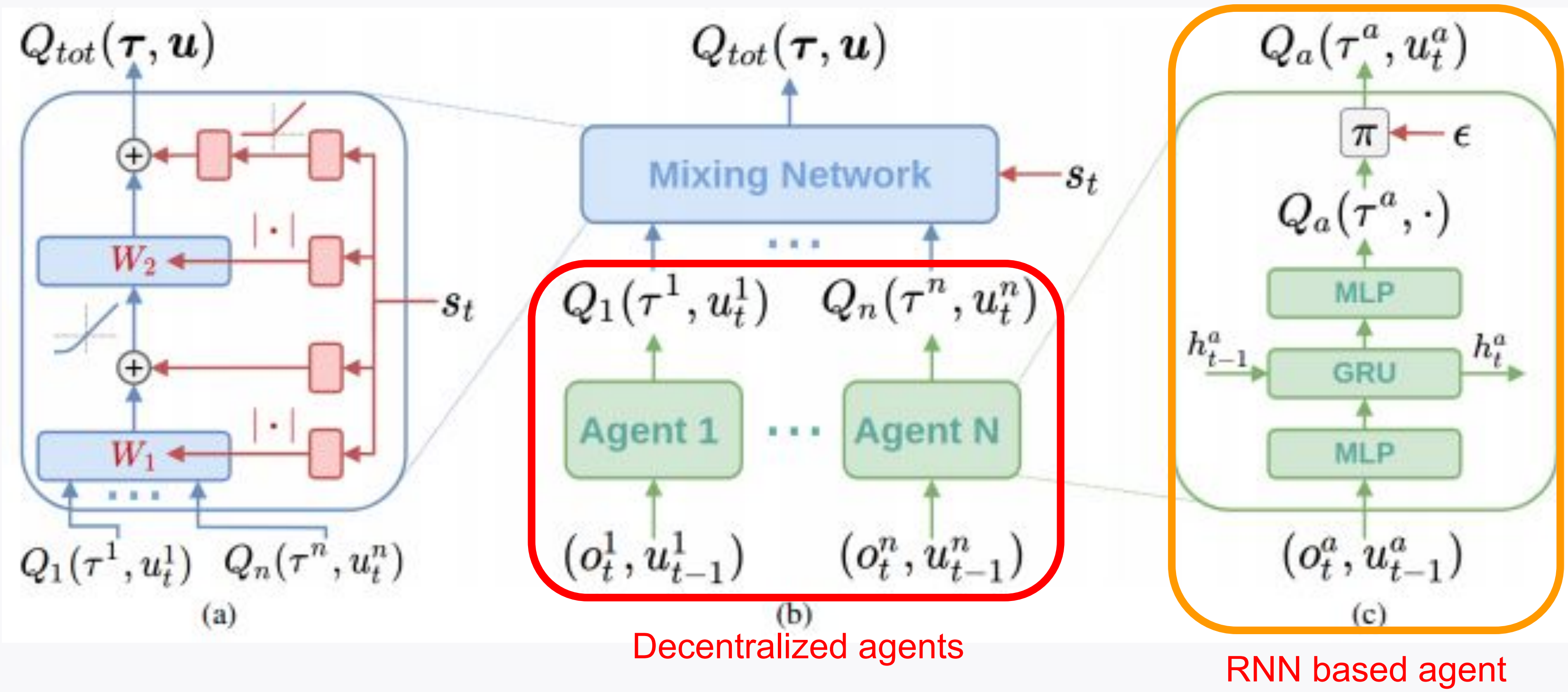
$$\operatorname*{argmax}_{\mathbf{u}} Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) = \begin{pmatrix} \operatorname{argmax}_{u^1} Q_1(\tau^1, u^1) \\ \vdots \\ \operatorname{argmax}_{u^n} Q_n(\tau^n, u^n) \end{pmatrix}$$

Full diagram of QMIX



(image source: Rashid et. al. 2018)

QMIX concept



(image source: Rashid et. al. 2018)

QMIX concept

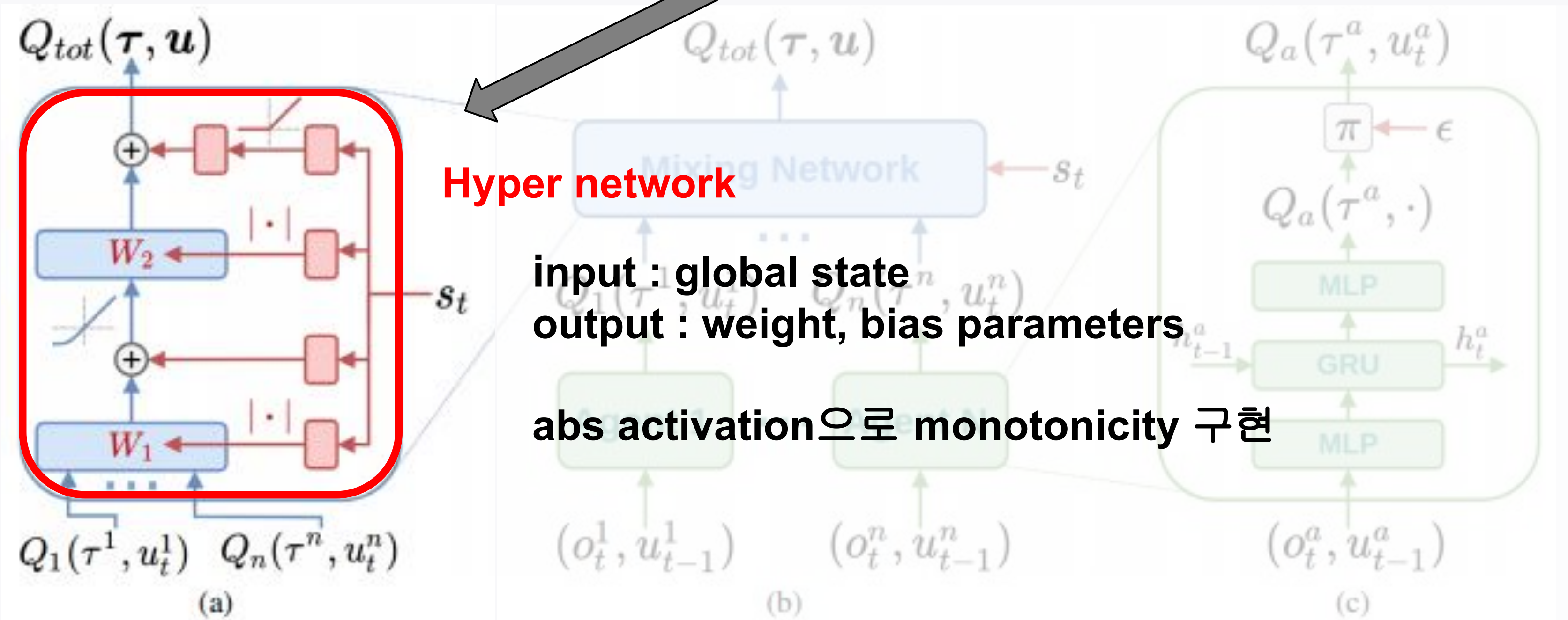
Monotonicity Q value among cooperative agents

$$\operatorname*{argmax}_{\mathbf{u}} Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) = \begin{pmatrix} \operatorname{argmax}_{u^1} Q_1(\tau^1, u^1) \\ \vdots \\ \operatorname{argmax}_{u^n} Q_n(\tau^n, u^n) \end{pmatrix}$$

하도록 Q function 학습

Q_a가 증가할 때 Q_{tot} 도 증가한다

y=Ax+b, A >= 0



Hyper network

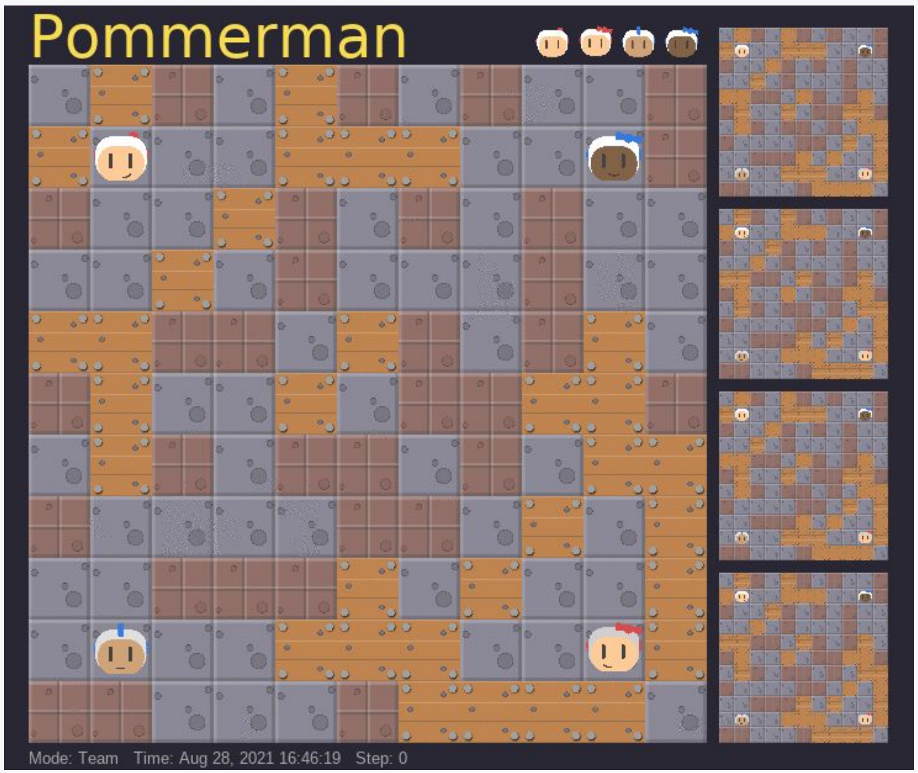
input : global state
output : weight, bias parameters

abs activation으로 monotonicity 구현

(image source: Rashid et. al. 2018)

QMIX result

Against peaceful agents



Against stopped agents

Future work

1. **Reward scheme modification**
   Sparse , Noisy reward problem
   Aggressive agents training

2. **Observation modification**
   Partially observable
   Full observable

3. **Imitation learning**
   Exploration problem (10^2400 >> 바둑)

4. **COMA, QMIX Success…!!!!!!**

Reference

Zhang, K., Yang, Z., & Başar, T. (2021). Multi-agent reinforcement learning: A selective overview of theories and algorithms. Handbook of Reinforcement Learning and Control, 321-384.

Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., & Whiteson, S. (2018, April). Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 32, No. 1).

Rashid, Tabish, et al. "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning." *International Conference on Machine Learning*. PMLR, 2018.