Multi-Agent Reinforcement Learning is A Sequence Modeling Problem (2022)

Muning Wen[1, 2], Jakub Grudzien Kuba[3], Runji Lin[4], Weinan Zhang[1], Ying Wen[1], Jun Wang[2, 5], Yaodong Yang[6, 7, *]
[1]Shanghai Jiao Tong University, [2]Digital Braing Lab, [3]University of Oxford, [5]University College London
[6]Beijing Institute for General AI, [7]Institute for AI, Peking University

IIE8557-01 동적계획법과 강화학습

경영과학연구실 김윤석

- 최근 몇 년 동안, Sequence model(SM)은 자연어 처리 분야에서 상당한 진전을 이룸
- SM은 언어뿐만 아니라 일반적으로 널리 적용 가능한 기본 모델임
- Computer vision에서는 이미지를 나누고 나누어진 이미지를 토큰처럼 순차적으로 사용하여 좋은 성능을 보임
- Transformer는 강화학습 커뮤니티에서 엄청난 관심을 받았으며 성공적인 결과를 보임
- 다중 에이전트 강화학습에서는 Transformer를 효과적으로 사용하지 못하고 있음
- 예를 들어 단순히 Transformer를 각 에이전트의 정책 신경망으로 사용한다고 해서 성능이 개선될 것임을 보장할 수 없음
- 다중 에이전트 강화학습에서는 강력한 SM (Transformer)가 활발히 제공되는 동안 그 이점을 효과적으로 활용하지 못하고 있음

MARL에서 효과적으로 Transformer를 사용하기 위한
모델링과 프레임워크를 연구함

다중 에이전트 이점 분해를 효과적으로
모델링하기 위한 Sequence model 사용을 연구함

- $\mathcal{N}$: 에이전트 집합
- $\mathcal{O}$: 각 에이전트의 관찰 공간을 모두 곱한 것, 이를 joint observation space라고 함
- $\mathcal{A}$: 에이전트들의 행동 공간의 곱, joint action space라고 함
- $\mathcal{R}$: $\mathcal{O} \times \mathcal{A} \to [-R_{max}, R_{max}]$: joint reward function으로 관찰과 행동을 입력으로 받아 보상을 출력함
- $\mathcal{P}$: $\mathcal{O} \times \mathcal{A} \times \mathcal{O} \to \mathbb{R}$: transition probability function
- $\gamma \in [0,1)$: discount factor

$$Q_{\boldsymbol{\pi}}(\boldsymbol{o}, \boldsymbol{a}) \triangleq \mathbb{E}_{\mathbf{o}_{1:\infty} \sim P, \mathbf{a}_{1:\infty} \sim \boldsymbol{\pi}} \left[R^{\gamma} | \mathbf{o}_0 = \boldsymbol{o}, \mathbf{a}_0 = \boldsymbol{a} \right], \tag{1}$$

$$V_{\boldsymbol{\pi}}(\boldsymbol{o}) \triangleq \mathbb{E}_{\mathbf{a}_0 \sim \boldsymbol{\pi}, \mathbf{o}_{1:\infty} \sim P, \mathbf{a}_{1:\infty} \sim \boldsymbol{\pi}} \left[R^{\gamma} | \mathbf{o}_0 = \boldsymbol{o} \right].$$

- 여러 에이전트들이 공유 보상을 받았을 때, 개별 에이전트가 팀의 성공 또는 실패에 얼마나 기여했는지를 판단하는 것은 어려움
- Multi-agent observation-value function(2)은 위 문제를 해결하기 위해 제안 됨
- 식 (2)는 m개의 다중 에이전트의 observation-value function임
- 본 이론은 기여도를 판단하기 위해 전체 에이전트 그룹에서 m(특정)그룹의 마진을 multi-agent advantage function(3)을 제안함
- 본 이론에서 에이전트의 순열 1:n이 주어졌다면, (가)식은 항상 성립함
- (가)식은 행동을 순차적으로 선택에 대한 직관을 제공함
- (가)식을 통해 행동을 선택하게 되면 전체 행동 공간에서 검색하지 않아 복잡도는 $\sum_{n=1}|A^j|$로 선형으로 증가함

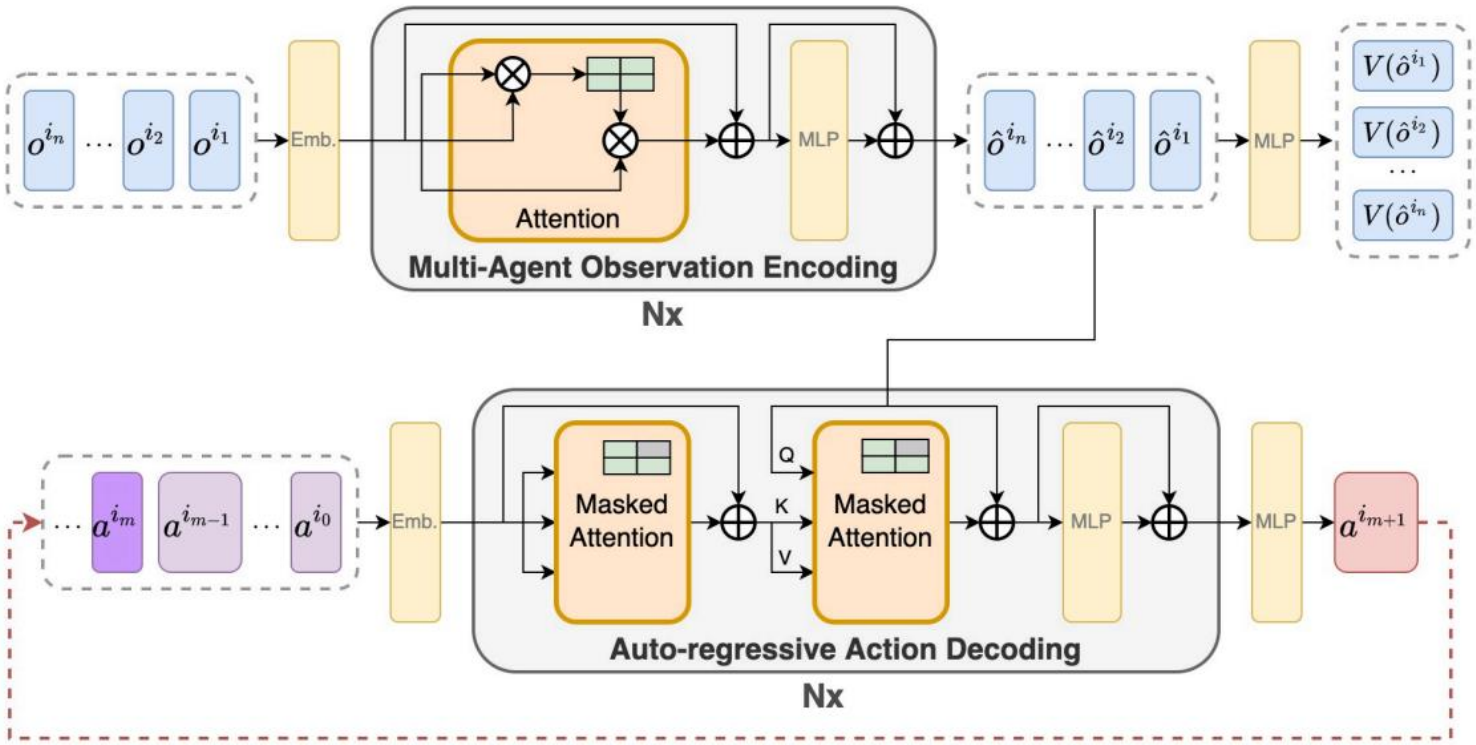
$$Q_{\boldsymbol{\pi}}(\boldsymbol{o}, \boldsymbol{a}^{i_{1:m}}) \triangleq \mathbb{E}\left[R^{\gamma} | \mathbf{o}_0^{i_{1:n}} = \boldsymbol{o}, \mathbf{a}_0^{i_{1:m}} = \boldsymbol{a}^{i_{1:m}}\right], \tag{2}$$

$$A_{\boldsymbol{\pi}}^{i_{1:m}}(\boldsymbol{o}, \boldsymbol{a}^{j_{1:h}}, \boldsymbol{a}^{i_{1:m}}) \triangleq Q_{\boldsymbol{\pi}}^{j_{1:h}, i_{1:m}}(\boldsymbol{o}, \boldsymbol{a}^{j_{1:h}}, \boldsymbol{a}^{i_{1:m}}) - Q_{\boldsymbol{\pi}}^{j_{1:h}}(\boldsymbol{o}, \boldsymbol{a}^{j_{1:h}}). \tag{3}$$

$$A_{\boldsymbol{\pi}}^{i_{1:n}}(\boldsymbol{o}, \boldsymbol{a}^{i_{1:n}}) = \sum_{m=1}^{n} A_{\boldsymbol{\pi}}^{i_m}(\boldsymbol{o}, \boldsymbol{a}^{i_{1:m-1}}, a^{i_m}). \tag{가}$$

- 각 에이전트의 관찰을 받아 임베딩 하여 Transformer의 encoder로 입력함
- 인코딩된 관찰은 디코더와 Value function 출력 layer에 입력으로 사용 됨
- 디코더는 action의 sequence modeling을 가능하게 하며, masked attention을 사용해 multi-agent advantage 값을 연산 가능하게 함

Network architecture

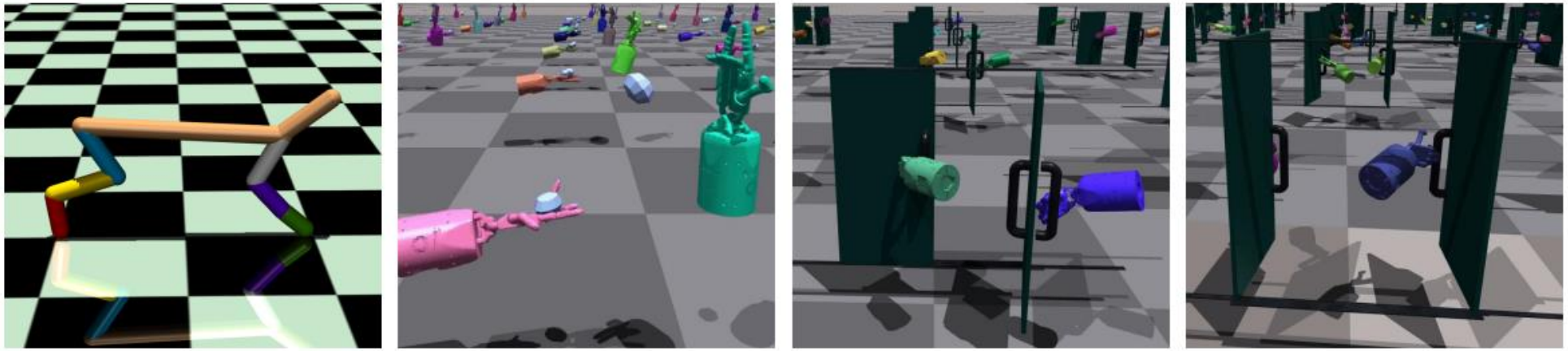


- 인코더는 가치 함수를 근사하도록 학습함, 이 손실함수는 Bellman error를 최소화함
- 디코더는 PPO와 같은 손실함수를 사용하여 학습함

$$L_{\text{Encoder}}(\phi) = \frac{1}{Tn}\sum_{m=1}^{n}\sum_{t=0}^{T-1}\left[R(\mathbf{o}_t, \mathbf{a}_t) + \gamma V_{\bar{\phi}}(\hat{\mathbf{o}}_{t+1}^{i_m}) - V_{\phi}(\hat{\mathbf{o}}_t^{i_m})\right]^2, \tag{4}$$

$$L_{\text{Decoder}}(\theta) = -\frac{1}{Tn}\sum_{m=1}^{n}\sum_{t=0}^{T-1}\min\left(\mathrm{r}_t^{i_m}(\theta)\hat{A}_t, \text{clip}(\mathrm{r}_t^{i_m}(\theta), 1\pm\epsilon)\hat{A}_t\right), \tag{5}$$

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



(a) HalfCheetah (b) CatchOver2Underarm (c) DoorOpenInward (d) DoorCloseOutward

Figure 3: Demonstrations of the Bi-DexHands and the HalfCheetah environments.

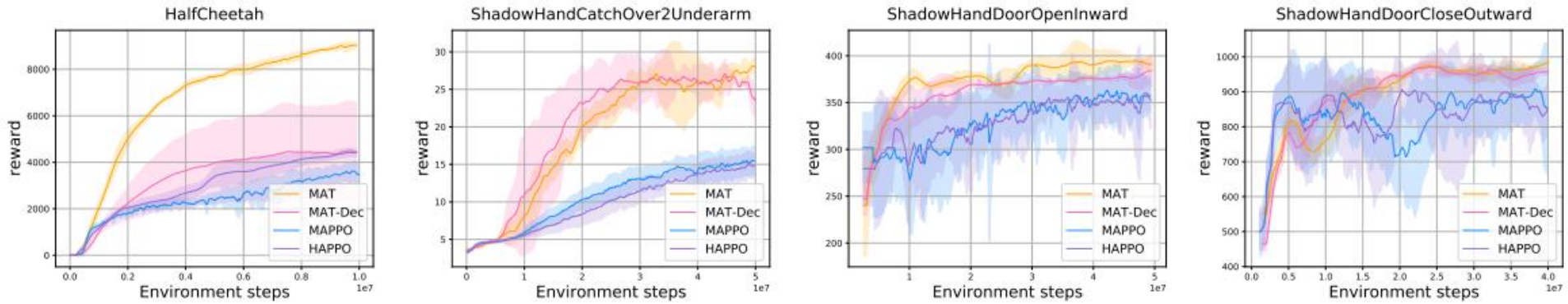


Figure 4: Performance comparisons on the Multi-Agent MuJoCo and the Bi-DexHands benchmarks.

Table 1: Performance evaluations of win rate and standard deviation on the SMAC benchmark, where UPDeT's official codebase supports several Marine-based tasks only.

Task	Difficulty	MAT	MAT-Dec	MAPPO	HAPPO	QMIX	UPDeT	Steps
3m	Easy	**100.0**(1.8)	**100.0**(1.1)	**100.0**(0.4)	**100.0**(1.2)	96.91.3	**100.0**(5.2)	5e5
8m	Easy	**100.0**(1.1)	97.5(2.5)	96.8(2.9)	97.5(1.1)	97.71.9	96.3(9.7)	1e6
1c3s5z	Easy	**100.0**(2.4)	**100.0**(0.4)	**100.0**(2.2)	97.5(1.8)	96.9(1.5)	/	2e6
MMM	Easy	**100.0**(2.2)	98.1(2.1)	95.6(4.5)	81.2(22.9)	91.2(3.2)	/	2e6
2c vs 64zg	Hard	**100.0**(1.3)	95.9(2.3)	**100.0**(2.7)	90.0(4.8)	90.3(4.0)	/	5e6
3s vs 5z	Hard	**100.0**(1.7)	**100.0**(1.3)	**100.0**(2.5)	91.9(5.3)	92.3(4.4)	/	5e6
3s5z	Hard	**100.0**(1.9)	**100.0**(3.3)	72.5(26.5)	90.0(3.5)	84.3(5.4)	/	3e6
5m vs 6m	Hard	**90.6**(4.4)	83.1(4.6)	88.2(6.2)	73.8(4.4)	75.8(3.7)	**90.6**(6.1)	1e7
8m vs 9m	Hard	**100.0**(3.1)	95.0(4.6)	93.8(3.5)	86.2(4.4)	92.6(4.0)	/	5e6
10m vs 11m	Hard	**100.0**(1.4)	**100.0**(2.0)	96.3(5.8)	77.5(9.7)	95.8(6.1)	/	5e6
25m	Hard	**100.0**(1.3)	86.9(5.6)	**100.0**(2.7)	0.6(0.8)	90.2(9.8)	2.8(3.1)	2e6
27m vs 30m	Hard+	**100.0**(0.7)	95.3(2.2)	93.1(3.2)	0.0(0.0)	39.2(8.8)	/	1e7
MMM2	Hard+	**93.8**(2.6)	91.2(5.3)	81.8(10.1)	0.3(0.4)	88.3(2.4)	/	1e7
6h vs 8z	Hard+	**98.8**(1.3)	93.8(4.7)	88.4(5.7)	0.0(0.0)	9.7(3.1)	/	1e7
3s5z vs 3s6z	Hard+	**96.5**(1.3)	85.3(7.5)	84.3(19.4)	82.8(21.2)	68.8(21.2)	/	2e7

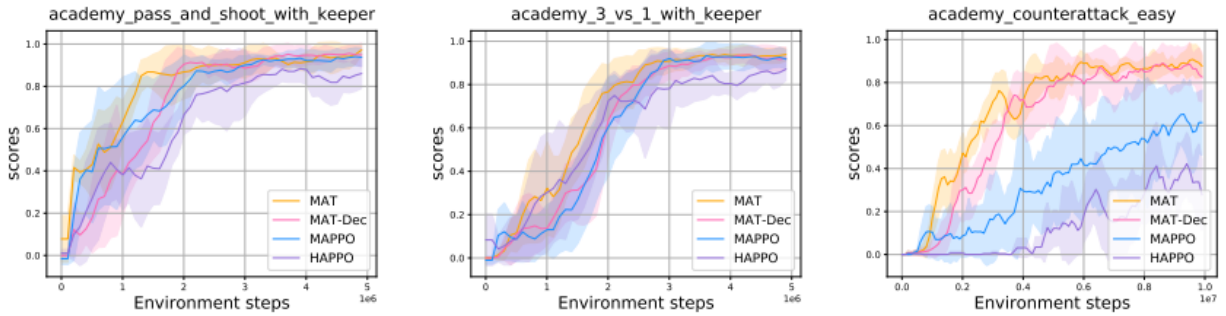


Figure 5: Performance comparison on the Google Research Football tasks with 2-4 agents from left to right respectively.

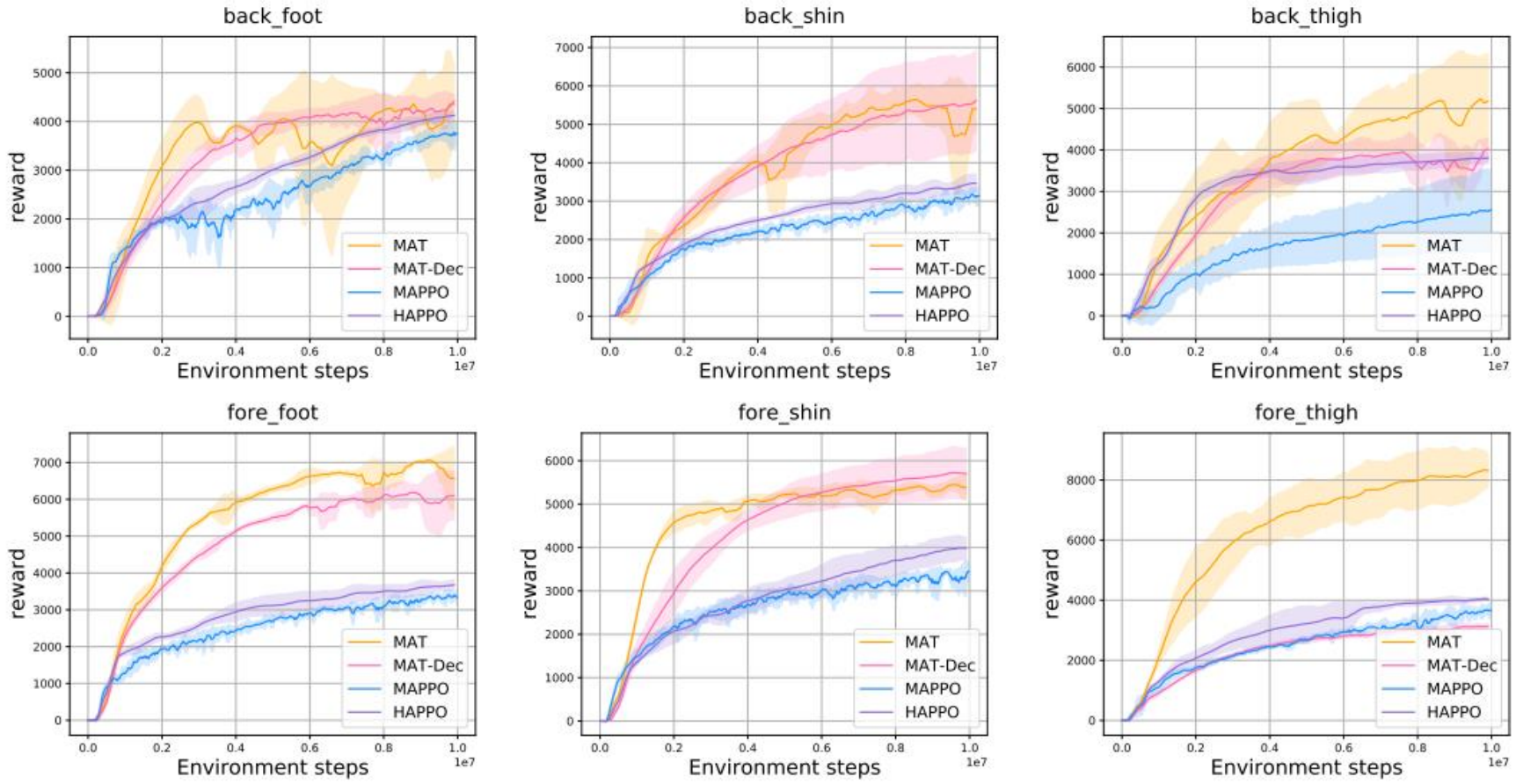


Figure 6: Performance on the HalfCheetah task with different disabled joints shown in Figure (3a).

Table 2: Median evaluation win rate and the standard deviation on the SMAC benchmark for pre-trained models with different number of online examples.

Methods	MAT				MAPPO				MAT-from scratch			
#examples	0%	1%	5%	10%	0%	1%	5%	10%	0%	1%	5%	10%
5m vs 6m	0.0(0.0)	0.0(0.0)	**5.8**(3.1)	18.8(7.1)	0.0(0.0)	0.0(0.0)	4.3(3.8)	**21.9**(12.2)	0.0(0.0)	0.0(0.0)	1.9(1.3)	3.8(2.1)
8m	100(0.0)	100(1.2)	100(0.3)	100(2.1)	100(0.0)	100(1.4)	100(0.3)	100(1.4)	0.0(0.0)	10.6(23.8)	92.5(3.7)	100(1.4)
27m vs 30m	0.0(0.0)	6.3(2.4)	**53.8**(16.4)	**71.2**(8.2)	**9.4**(3.6)	**15**(5.9)	26.2(7.8)	26.8(9.7)	0.0(0.0)	0.0(0.0)	0.0(0.3)	0.3(15.6)
2s vs 1sc	0.0(0.0)	15.6(13.8)	100(9.7)	100(0.0)	0.0(0.0)	**43.1**(17.6)	100(1.1)	100(1.8)	0.0(0.0)	19.3(33.3)	96.3(6.2)	100(0.3)
1c3s5z	**3.1**(1.8)	**5.6**(5.0)	**82.5**(5.5)	**100**(2.7)	**3.1**(1.8)	4.3(4.9)	73.8(13.0)	97.5(2.1)	0.0(0.0)	7.5(4.8)	87.5(3.9)	100(1.4)
MMM2	0.0(3.6)	0.0(1.8)	**33.8**(13.7)	**62.5**(12.1)	0.0(0.0)	0.0(1.4)	13.8(7.0)	36.2(9.6)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.7)

Table 3: Average evaluation score and standard deviation on Multi-Agent MuJoCo for pre-trained models with different number of online examples.

Methods	MAT				MAPPO				MAT-from scratch			
#examples	0%	1%	5%	10%	0%	1%	5%	10%	0%	1%	5%	10%
back foot	2100(89)	2837(95)	**4691**(235)	**5646**(79)	**2936**(301)	**3017**(135)	3221(119)	3304(129)	-0.44(0.4)	-5.18(11)	670(1098)	1635(1184)
back shin	**4005**(316)	**4143**(230)	**6077**(209)	**7176**(74)	2406(32)	2542(108)	2796(137)	2955(127)	-0.31(0.1)	-3.95(17)	743(537)	1252(1123)
back thigh	**5361**(45)	**5641**(150)	**7101**(119)	**7460**(61)	3043(79)	3060(143)	3217(33)	3353(71)	-0.54(0.3)	-4.87(7.7)	930(589)	2067(861)
fore foot	**1313**(512)	**1955**(232)	**4856**(146)	**6054**(172)	623(44)	970(185)	2025(371)	2480(239)	-0.37(0.2)	-2.25(7.9)	1821(157)	2877(106)
fore shin	**2435**(13)	**2617**(71)	**3851**(57)	**4373**(83)	1715(55)	2457(125)	3096(59)	3310(54)	-0.15(0.06)	-0.96(6.0)	1461(101)	3003(316)
fore thigh	**5631**(321)	**6448**(417)	**7952**(109)	**8347**(81)	3087(110)	3171(83)	3340(52)	3519(59)	-0.29(0.3)	0.82(14)	1021(177)	2600(215)

Conclusions

- 본 연구는 Sequence model의 MARL에 효과적으로 사용함
- 본 연구는 Sequence modeling을 time step을 기준으로 한게 아닌 agent의 action 선택을 기준으로 한 점에서 기여가 크다고 생각함
- 본 연구는 Action 선택을 Sequence modeling 하여 multi-agent advantage decomposition theorem을 효과적으로 적용함

Q & A