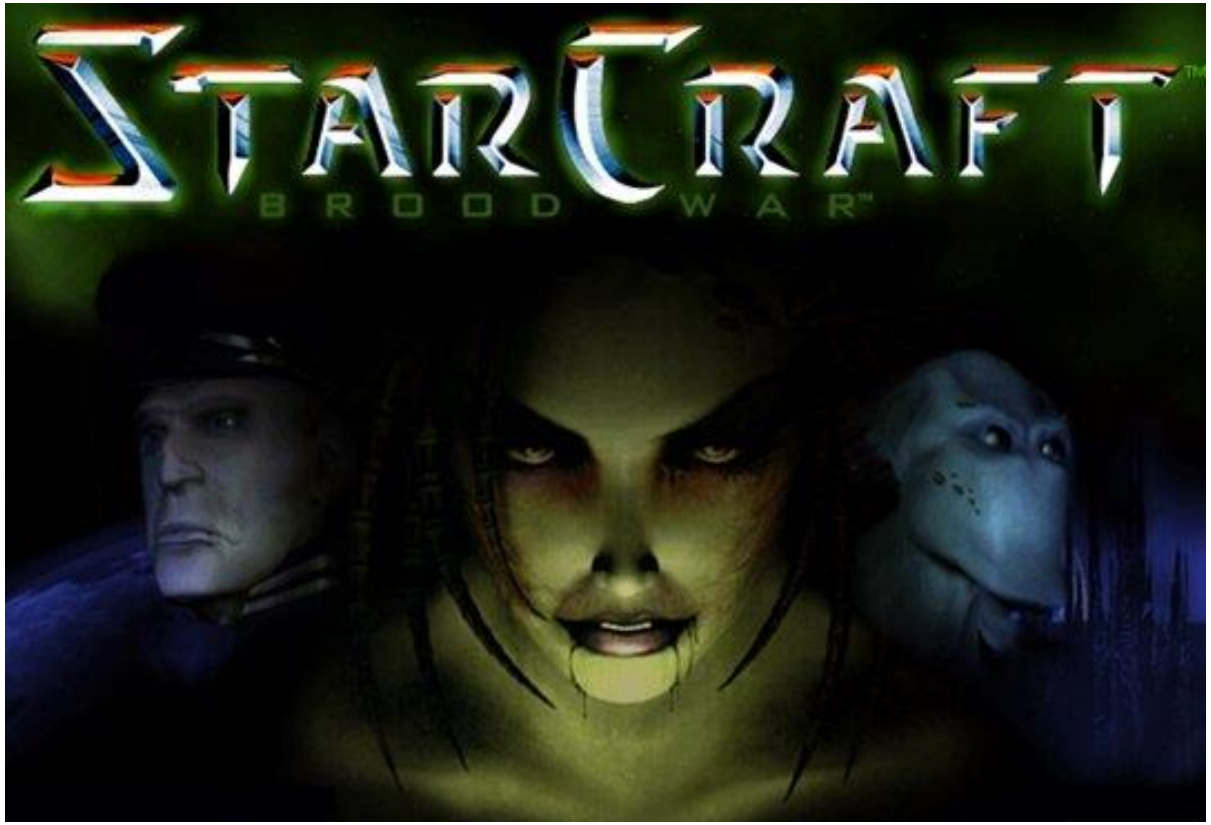
Bidirectionally-Coordinated Nets Emergence of Human-level Coordination in Learning to Play StarCraft Combat Games (2017)

Peng, Peng, et al. University College London, Alibaba Group arXiv preprint arXiv:1703.10069 (2017)

IIE8557-01 고등강화학습

경영과학연구실 이태헌

전략 시뮬레이션 게임 Starcraft



Introduction

- Single-Agent 게임들에서는 AI가 사람을 앞지름 (Atari, 바둑, Texas Holdem 등)
- 사회에서의 지식은 집단 상호 작용으로 만들어짐. 이를 Artificial General Intelligence에도 적용하고자 함
- Real-Time Strategy (RTS) game인 Starcraft가 효과적
- Agent 수가 증가함에 따라 파라미터 공간이 기하급수적으로 증가함. 대규모 Multi-agent 학습에서는 Challenge한 문제

‘Starcraft를 실험 시나리오로 사용, Multi-agent들이 한 팀으로 행동하여 게임을 승리하고자 함’

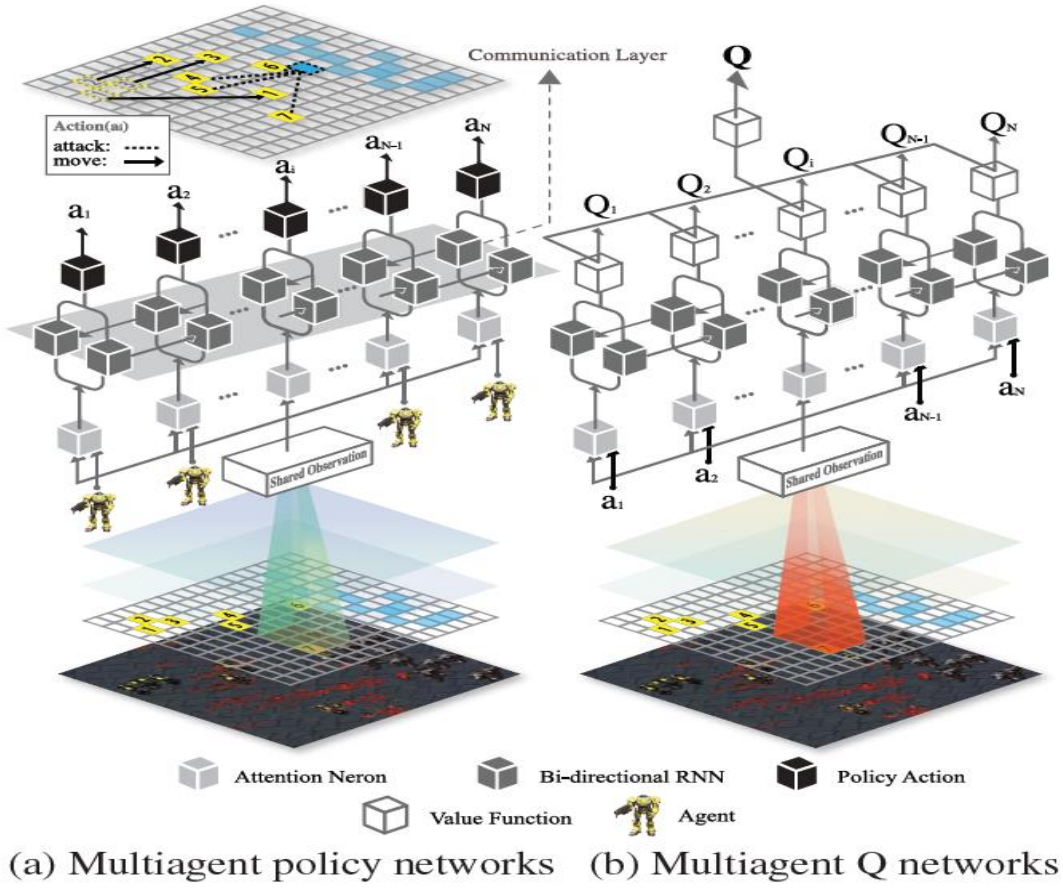
- 한 팀으로 행동하기 위해서는 Communication이 중요함. Starcraft 같은 RTS 게임에서는 확장 가능하며 효과적인 Communication protocol이 필요함

Multi-agent bidirectionally-coordinated network (BiCNet) with a vectorized extension of actor-critic formulation

- 각각의 agent들이 BiCNet을 통해 Communication
- Evaluation-decision-making process 통해 학습
- Parameter 공유와 Dynamic grouping으로 확장성 문제 해결 시도
- 사람이 게임하거나 label data가 없어도 BiCNet은 여러 종류들로 이루어진 Agent 역할을 학습할 수 있음

Bidirectionally-Coordinated Net (BiCNet)

- Bi-direction recurrent network가 개별 agent policy 및 Q-network를 연결하는데 사용됨
- Multi-agent deterministic actor-critic 통해 학습됨



(a) Multiagent policy networks (b) Multiagent Q networks

- **Bi-directional RNN** : Agent 사이 정보 교환과 협력을 향상시킴. 순방향, 역방향으로 함께 데이터 처리하여 agent의 state, action 인코딩
- Policy network, Q-network 모두 Bi-directional RNN 구조를 기반으로 함
- Policy network는 shared observation (모든 agent들이 관찰할 수 있는 공통의 정보 및 데이터) 과 함께 지역적 관점을 입력으로 받아, 각 개별 agent 행동 반환
- **Communication layer :** Multi-agent에서 agent 간 정보 교환을 위해 사용되는 network layer. 정보 전달, 가중치 공유 역할을 함
- 각 agent는 두 network의 파라미터들을 공유. 다수 agent들이 경험한 다양한 상황을 더 빨리 학습할 수 있음

MDP Modeling

$$(\mathcal{S}, \{\mathcal{A}_i\}_{i=1}^{N}, \{\mathcal{B}_i\}_{i=1}^{M}, \mathcal{T}, \{\mathcal{R}_i\}_{i=1}^{N+M})$$

- S : 모든 Agent들이 공유하는 현재 게임의 state space
- A_i : Controller agent i action space, i ∈ $[1, N]$
- B_j : Enemy j action space, j ∈ $[1, M]$
- T : S x A^N x B^M -> S Environment deterministic transition function
- R_i : : S x A^N x B^M -> R agent/enemy i reward function, i ∈ $[1, N + M]$

- 단순화하기 위해 모든 agent(controller, enemy)들이 같은 action space를 공유하는 것으로 가정

Reward function

$$r(\mathbf{s},\mathbf{a},\mathbf{b}) \equiv \frac{1}{M}\sum_{j=N+1}^{N+M}\Delta\mathcal{R}_j^t(\mathbf{s},\mathbf{a},\mathbf{b}) - \frac{1}{N}\sum_{i=1}^{N}\Delta\mathcal{R}_i^t(\mathbf{s},\mathbf{a},\mathbf{b}) \quad (1)$$

- Deterministic policy a_θ : S -> A^N (controlled agents)
- Deterministic policy $b_\emptyset$: S -> B^M (enemies)

- Eq. (1)은 Controlled agent 관점. Enemy의 global reward는 정확히 반대 (zero-sum game)
  (Global reward : 같은 팀에 각 agent들은 같은 reward를 공유)

- Controlled agent 체력에 비해 Enemy 체력을 최대한 시키는 것이 필요. 이를 최대화하는 reward

$$r_i(\mathbf{s},\mathbf{a},\mathbf{b}) \equiv \frac{1}{|j|}\sum_{j=N+1\cap top\text{-}K(i)}^{M}\Delta\mathcal{R}_j(\mathbf{s},\mathbf{a},\mathbf{b}) - \frac{1}{|i'|}\sum_{i'=1\cap top\text{-}K(i)}^{N}\Delta\mathcal{R}_{i'}(\mathbf{s},\mathbf{a},\mathbf{b})$$

- Eq (1) 각 agent의 local reward와 다른 agent들과의 상호작용 영향 추가
- 각 agent i가 유지하고 있는 top-K(i)
- 협업이 가능하게 Controlled agent, enemy 상관없이 k개 유닛들의 체력 변화를 reward

Minimax Q-learning

$$Q^*_{\mathrm{SG}}(\mathbf{s}, \mathbf{a}, \mathbf{b}) = r(\mathbf{s}, \mathbf{a}, \mathbf{b}) + \lambda \max_{\theta} \min_{\phi} Q^*_{\mathrm{SG}}(\mathbf{s}', \mathbf{a}_\theta(\mathbf{s}'), \mathbf{b}_\phi(\mathbf{s}')) \quad (2)$$

- Controlled agent 목적은 expected sum of discounted rewards를 최대화하는 policy를 학습
- Enemy의 joint policy는 expected sum을 최소화하는 것

Simpler MDP problem

$$Q^*_{\mathrm{SG}}(\mathbf{s}, \mathbf{a}, \mathbf{b}) = r(\mathbf{s}, \mathbf{a}, \mathbf{b}) + \lambda \max_{\theta} \min_{\phi} Q^*_{\mathrm{SG}}(\mathbf{s}', \mathbf{a}_\theta(\mathbf{s}'), \mathbf{b}_\phi(\mathbf{s}')) \quad (2)$$

$$Q^\theta(\mathbf{s}, \mathbf{a}) = r(\mathbf{s}, \mathbf{a}) + \lambda Q^\theta(\mathbf{s}', \mathbf{a}_\theta(\mathbf{s}')) \quad (3)$$

where we drop notation $\mathbf{b}_\phi$ for brevity.

- Enemies의 policy를 학습하고 고정시키면, Eq. (2)에서 정의된 Stochastic game은 더 간단한 MDP 문제로 바뀜

Policy network (Actor), Q-network (Critic)

$$\nabla_\theta J(\theta) = \frac{\partial}{\partial \theta} \int_{\mathcal{S}} p_1(\mathbf{s}) \sum_{i=1}^{N} Q_i^{\mathbf{a}_\theta}(\mathbf{s}, \mathbf{a}) \mathrm{d}\mathbf{s} = \mathbb{E}_{\mathbf{s} \sim \rho_{\mathbf{a}_\theta}^{\mathcal{T}}(\mathbf{s})} \left[\sum_{j}^{N} \left(\sum_{i=1}^{N} \frac{\partial Q_i^{\mathbf{a}_\theta}(\mathbf{s}, \mathbf{a})|_{\mathbf{a}=\mathbf{a}_\theta(\mathbf{s})}}{\partial \mathbf{a}_j} \right) \frac{\partial \mathbf{a}_{j,\theta}(\mathbf{s})}{\partial \theta} \right]$$

$$\nabla_\xi L(\xi) = \mathbb{E}_{\mathbf{s} \sim \rho_{\mathbf{a}_\theta}^{\mathcal{T}}(\mathbf{s})} \left[\sum_{i=1}^{N} \left(r_i(\mathbf{s}, \mathbf{a}_\theta(\mathbf{s})) + \lambda Q^\xi(\mathbf{s}', \mathbf{a}_\theta(\mathbf{s}')) - Q^\xi(\mathbf{s}, \mathbf{a}_\theta(\mathbf{s})) \right) \frac{\partial Q_i^\xi(\mathbf{s}, \mathbf{a}_\theta(\mathbf{s}))}{\partial \xi} \right]$$

- Off-policy deterministic actor-critic
- Actor, Critic network 모두 SGD 통해 업데이트
- 모든 유닛을 다 고려하는 전체적 관점에서 계산 하고 backpropagate 시 유닛 별 모델과 전체 모델 모두 적용

실험 설계

- Easy combats
  - 3 Marines vs 1 Super Zergling
  - 3 Wraiths vs 3 Mutalisks

- Difficult combats
  - 5 Marines vs 5 Marines
  - 15 Marines vs 16 Marines
  - 20 Marines vs 30 Zerglings
  - 10 Marines vs 13 Zerglings
  - 15 Wraiths vs 17 Wraiths

- Heterogeneous combats
  - 2 Dropships and 2 Tanks vs 1 Ultralisk

Parameter tuning

- 800회 에피소드에서 학습된 BiCnet 모델 선택하여 100회의 독립적인 게임에서 테스트
- 배치 32를 가진 모델은 600k training step 후 가장 높은 승률과 가장 높은 Mean Q-value 값을 얻음

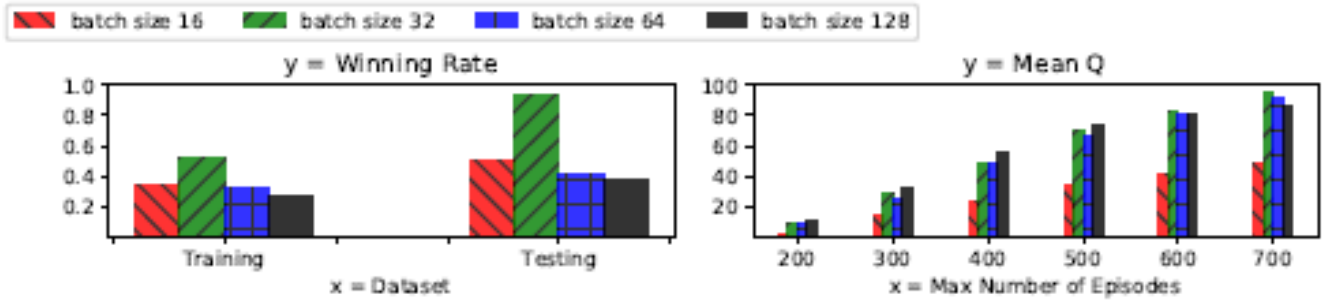


Figure 2: The impact of **batch_size** in combat *2 Marines vs. 1 Super Zergling*.

Parameter tuning

- 학습 에피소드 수에 대한 승률을 시각화하여 파라미터 학습 수렴 속도 비교
- BiCNet 모델이 baseline 모델들 보다 빠르게 수렴

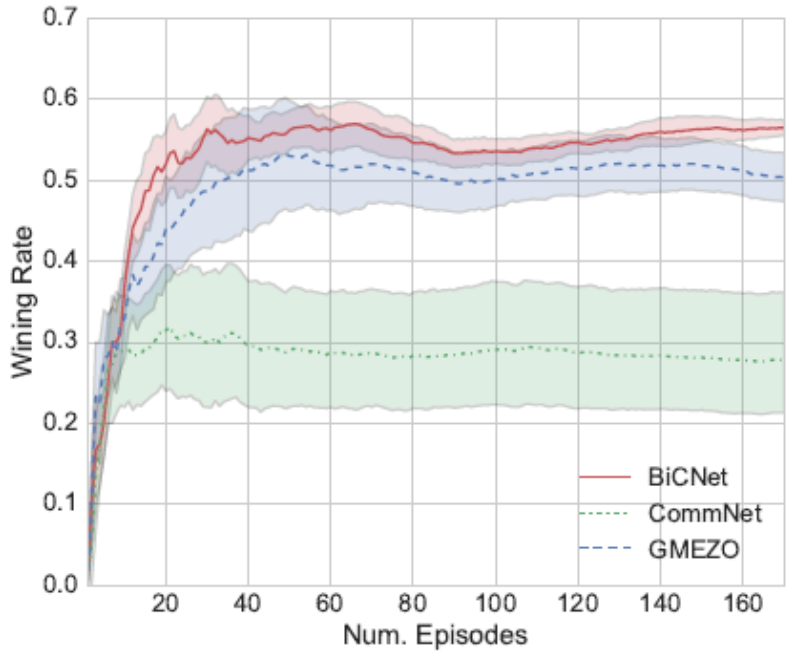


Figure 3: Learning Curves in Combat "10 *Marines* vs. 13 *Zerglings*"

- CommNet : 여러 agent들이 소통하는 것을 학습하기 위한 multi-agent network
- GMEZO : Greedy MDP with Episodic Zero-order optimization

BiCNet가 baseline 모델 평균 승률 비교

- 5개 시나리오 중 4개에서 BicNET은 다른 Baseline 모델들 능가함
- Agent 수가 10 이상일 때, 성능 차이가 더 크게 발생 (Agent 간 많은 협력이 필요한 상황에서 더 성능 좋음)

Table 1: Performance comparison. M: *Marine*, Z: *Zergling*, W: *Wraith*.

Combat	Rule Based			RL Based				
	Built-in	Weakest	Closest	IND	FC	GMEZO	CommNet	BiCNet
20 M vs. 30 Z	**1.00**	.000	.870	.940	.001	.880	**1.00**	**1.00**
5 M vs. 5 M	.720	.900	.700	.310	.080	.910	**.950**	.920
15 M vs. 16 M	.610	.000	.670	.590	.440	.630	.680	**.710**
10 M vs. 13 Z	.550	.230	.410	.522	.430	.570	.440	**.640**
15 W vs. 17 W	.440	.000	.300	.310	.460	.420	.470	**.530**

- Independent controller (IND) : 각 agent를 각각 전투에 컨트롤 (정보 공유 x)
- Fully-connected (FC) : agent간 소통 fully-connected
- CommNet : 여러 agent들이 소통하는 것을 학습하기 위한 multi-agent network
- GMEZO : Greedy MDP with Episodic Zero-order optimization

결과시각화

- 왼쪽 상단 : state with high Q value
- 오른쪽 하단 : state with low Q value
- Agent가 적과의 상호작용에서 행동에 따른 Q value 시각화
- t-SNE (고차원 데이터->저차원 공간 임베딩 알고리즘) 유사한 객체들은 가까운 거리에, 다른 객체들은 머리 떨어진 거리에 mapping

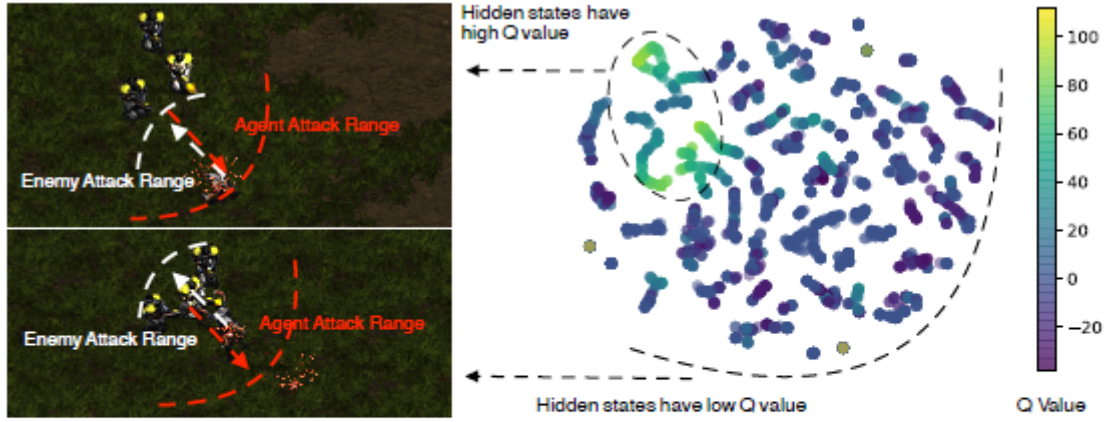
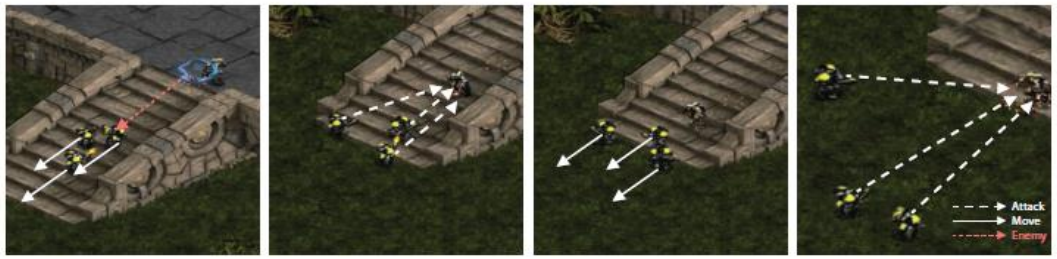


Figure 4: Visualisation for 3 Marines vs. 1 Super Zergling combat. **Upper Left**: State with high Q value; **Lower Left**: State with low Q value; **Right**: Visualisation of hidden layer outputs for each step using TSNE, coloured by Q values.

실험 설계

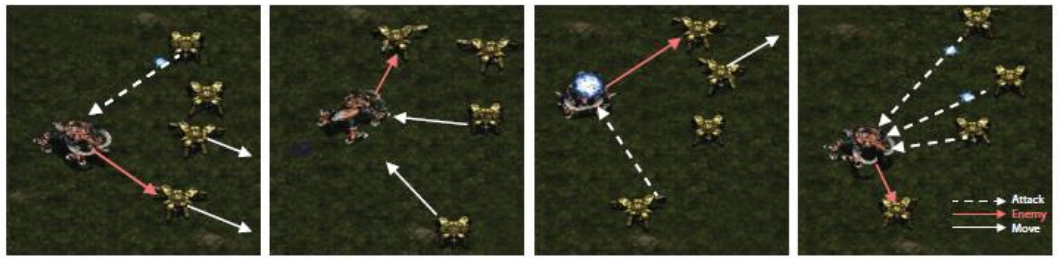
- Move without collision
    - 실험에서 가장 기본적인 task. 충돌하지 않고 움직이기.
- Hit and run
    - 기본적인 task. 공격하고 도망가기.
- Cover attack
    - 수준높은 task. 엄호하기.
- Focus fire without overkill
    - 수준높은 task. 필요한 만큼 공격하기.
- Collaboration between heterogeneous agents
    - 수준높은 task. 다른 특성의 유닛들이 협력하기.

실험 설계



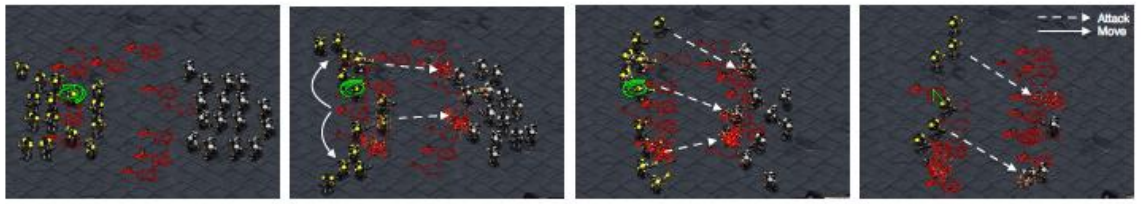
(a) time step 1 (b) time step 2 (c) time step 3 (d) time step 4

Figure 7: *Hit and Run* tactics in combat *3 Marines (ours) vs. 1 Zealot (enemy).*



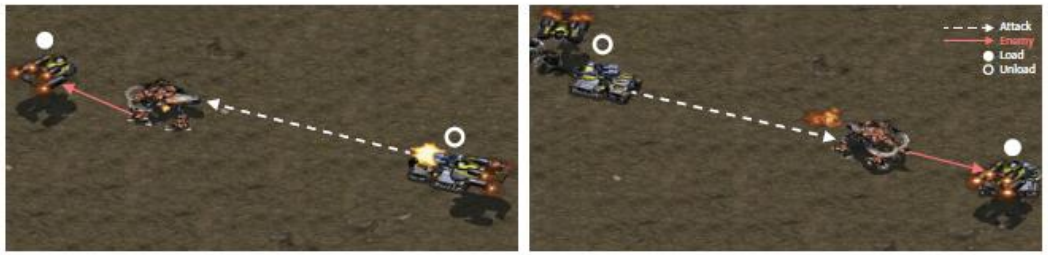
(a) time step 1 (b) time step 2 (c) time step 3 (d) time step 4

Figure 8: Coordinated cover attacks in combat *4 Dragoons (ours) vs. 1 Ultralisks (enemy)*



(a) time step 1 (b) time step 2 (c) time step 3 (d) time step 4

Figure 9: "focus fire" in combat *15 Marines (ours) vs. 16 Marines (enemy).*



(a) time step 1 (b) time step 2

Figure 10: Coordinated heterogeneous agents in combat 2 *Dropships* and 2 *tanks* vs. 1 *Ultralisk*.

Conclusions

- 새로운 딥러닝 기반의 멀티에이전트 강화 학습 방법 BiCNET 제안
- 각 Agent의 행동을 벡터화된 액터-크리틱 구조를 통해 학습하며, 에이전트간의 협력은 Bi-directional RNN 통해 이루어짐
- BiCNet은 종단간 학습 통해 여러 가지 효과적인 협력 전략을 습득할 수 있음을 보여줌
- 그러나, 게임 내에서의 협력의 복잡성을 정량적으로 측정하는 것은 어려운 문제. 향후 AI가 실제 플레이어와 경쟁하는 실험을 계획하며, 에이전트간 복잡한 상황에서의 정책 전달 방법이나 StarCraft 내에서 특정한 통신 언어의 발생 가능성 등을 더 깊게 연구할 예정

Q & A