Journal of the KIMST, Vol. 26, No. 4, pp. 361-370, 2023
ISSN 1598-9127(Print) · 2636-0640(Online)
DOI https://doi.org/10.9766/KIMST.2023.26.4.361

Research Paper 제어·구동 부문

# 전이학습을 활용한 군집제어용 강화학습의 효율 향상 방안에 관한 연구

이슬기[*,1) ] · 김권일[1)] · 윤석민[2)]

[1)] ㈜씨피프티원 기술연구소
[2)] 국방과학연구소 국방첨단과학기술연구원 인공지능자율센터

# Study on Enhancing Training Efficiency of MARL for Swarm Using Transfer Learning

Seulgi Yi[*,1)] · Kwon-Il Kim[1)] · Sukmin Yoon[2)]

[1)] *C51 Inc., R&D Department, Korea*
[2)] *AI Autonomy Technology Center, Advanced Defense Science & Technology Research Institute, Agency for Defense Development, Korea*

(Received 9 March 2023 / Revised 12 June 2023 / Accepted 13 July 2023)

## Abstract

Swarm has recently become a critical component of offensive and defensive systems. Multi-agent reinforcement learning(MARL) empowers swarm systems to handle a wide range of scenarios. However, the main challenge lies in MARL's scalability issue - as the number of agents increases, the performance of the learning decreases. In this study, transfer learning is applied to advanced MARL algorithm to resolve the scalability issue. Validation results show that the training efficiency has significantly improved, reducing computational time by 31 %.

Key Words : Multi-agent Reinforcement Learning(다개체 강화학습), Transfer Learning(전이학습), QMIX

## 기 호 설 명

$n$ : the number of agents
$a$ : actions
$s$ : states
$o$ : observations
$r$ : rewards
$\tau$ : action-observation history
$\gamma$ : discount factor
$N$ : index set of agents = $\{1,\cdots,n\}$
$N^i$ : index set of neighbors of agent i
$A$ : action space
$S$ : state space
$O$ : observation space
$R$ : returns

* Corresponding author, E-mail: ysg@c51.ai

$T$ : action-observation history space
$P$ : state transition function
$\pi$ : policy
$Q$ : state-action value functions
$p_x$, $p_y$ : positions in x, y-directions
$a_x$, $a_y$ : actions in x, y-directions
$hp$ : remaining health points

## 1. Introduction

최근 들어 무인기가 대통령 관저 근방 비행금지 구역을 침범[1]하거나, 석유 시설을 공격[2]하는 등 작은 사이즈나 저공비행 특성으로 인해 기존 대공 감시시스템의 취약점이 드러나는 사건 들이 빈번히 발생하고 있다. 특히, 무인기를 여러 대 운용하는 군집 시스템의 경우 일부 무인기가 낙오하더라도 최종적으로 임무를 완수할 수 있기 때문에 하드웨어/소프트웨어의 성능 발전 및 원가 하락에 힘입어 공격, 방어, 감시, 탐색 및 구조 등의 임무에 활용하는 방안에 관한 연구가 활발히 진행되고 있다[3-8]. 기존에 군집 제어를 위해 사용되던 행동 기반 알고리즘(behavior-based algorithm)[9]의 경우 예상치 못한 장애물이 나타나거나 일부 개체가 파손되는 등의 상황 변화에 능동적으로 대응하기 어렵다는 한계가 존재하기 때문에[10] 인공지능 기반의 강화학습(Reinforcement Learning, RL)을 적용하는 추세이다. 군집 시스템에 강화학습을 적용한 연구에는 다수의 적군으로부터 아군을 지키는 포식자-먹이(predator-prey) 문제를 풀기 위해 분산화된 개별 네트워크와 중앙화된 공동행동 가치함수(joint action value function)를 학습하는 MADDPG알고리즘[4], 다수 개체의 상태(states)를 mean embedding으로 표현하여 단일 네트워크처럼 학습하는 알고리즘[11], 이전 세대의 학습 모델로 제어되는 적군과 N 대 N 근접전(dogfight)을 통해 진화해나가도록 하는 self-play 알고리즘[12]을 적용하거나, 밀집된 군중으로부터 목표 대상을 호위하기 위한 임무를 위해 직접적으로 공동 행동 정책(joint action policy)을 학습하도록 한 사례[6] 등이 있다. 사람들의 경기 내용에서 배운 내용과 MARL을 통해 고난이도 전략게임인 Starcraft II에서 사람을 대상으로 이기도록 학습된 alphaStar[5] 또한 대표적인 사례 중 하나이다. 하지만, 다양한 연구 사례에도 불구하고 MARL이 가지고 있는 고질적인 문제점은 다수의 개체에 대해 강화 학습을 적용할 경우 각 개체의 개별적인 이득과 군집 전체의 이득이 반드시 부합하지만은 않을 수 있기 때문에 해의 변동성이 커질 수 있고, 환경에 의한 영향뿐 아니라 다른 개체와의 상호작용 또한 고려해야 하는 인자가 되기 때문에 군집의 규모가 커질수록 그 해를 구하는 것이 어렵거나 불가능해질 수 있다는 점이다[13].

MARL의 성능을 높이기 위해서는 군집의 규모와 관계없이 유효한 해를 얻을 수 있도록 하는 확장성(scalability)에 대한 고려가 필요하며, 이를 다루기 위해 대표적으로 다음의 세 가지 방법이 있다.

1) Embedding for multi-agent : 현재 군집의 상태를 표현하기 위해 각 개체의 상태 벡터를 모두 연결(concatenate)하는 경우 군집의 규모가 커질수록 탐색할 공간 또한 지수적으로 증가하여 적절한 해를 찾는데 막대한 자원이 들어가거나 아예 해를 찾지 못하게 되는 차원의 저주(curse of dimensionality) 문제가 발생한다. 이를 해결하기 위한 방법으로 군집 객체에 대한 상태를 표현하는 임베딩(embedding) 기법들이 개발되었으며, 단순 평균값, 임베딩된 벡터의 평균값[11], 어텐션(attention)으로 가중을 준 임베딩[14] 등 다양한 방법이 존재한다.
2) Transfer learning[15] : 확장성 문제를 해결하기 위한 다른 접근법으로 개체수에 상관없이 유사한 행동 패턴을 보이기도 한다는 점 혹은 단순한 임무의 조합으로 복잡한 임무의 구성이 가능하다는 점에 착안하여, 작은 규모의 환경 혹은 난이도가 낮은 임무에 대해 학습한 후 점진적으로 개체수를 늘리거나 문제의 복잡도를 높이는 방식을 사용할 수 있다. 전 단계에서 학습한 내용을 다음 단계로 전달하기 때문에 이러한 접근법을 전이 학습(transfer learning) 혹은 커리큘럼 학습(curriculum learning)이라고 한다.
3) Decentralized acting policy with centralized training: 개별 네트워크처럼 학습하여 개체 간의 상호작용이 전혀 고려되지 않거나(Independent Q-Learning, IQL), 모든 개체들의 상태 및 행동을 한꺼번에 고려한 통합 네트워크를 학습하여 군집 측면에서 개체의 행동이 결정될 수 있지만 차원의 저주문제를 지니게 되는 (Counterfactual Multi-Agent policy gradient, COMA)의 극단적인 두 방식을 적절히 혼용하여 개별 네트워크는 분산화하여 각자 정책(policy)대로 행

동하고, 통합 네트워크에서 이를 취합하여 학습하도록 함으로써 학습 성능과 효율을 확보할 수 있도록 하는 방식으로 QMIX[16], VDN[17], MADDPG[4] 등의 알고리즘이 개발되었다.

본 연구에서는 군집을 이루는 개체수가 증가할 때 발생하는 확장성 문제를 해결하고자 다수의 개체에 대한 학습 성능을 높일 수 있는 QMIX 알고리즘을 기반으로 하여, 소수의 개체에 대한 학습으로 얻은 지식을 다수의 개체를 제어하는 강화학습 모델에 전달하도록 하는 전이 학습을 도입하였다. 이를 이동하는 목표물을 방어 혹은 공격하기 위해 목표물에 접근하는 (migration) 문제에 적용하여 전이 학습이 MARL에서의 확장성 문제 해결에 실마리를 줄 수 있다는 점을 확인하였으며, 학습의 성능을 판단하기 어려운 강화학습에서 민감도 분석(sensitivity analysis)을 통해 현재 상태에서 특정 행동을 하게 된 근거에 대해 분석해 보았다. 본 논문은 2절에서 MARL에 관련된 배경지식을 소개하고, 3절에서는 본 연구에서 도입한 기법 들을 기술한 후, 4절과 5절에서 검증을 위한 문제 설정 및 결과와 분석 내용을 각각 다루도록 구성하였다.

## 2. Backgrounds

### 2.1 Swarm Markov Decision Process

군집 시스템은 Hüttenrauch[11]의 연구에서 기술된 바와 같이 현재 상태는 바로 이전 상태에 의해서만 결정된다는 마르코프 결정과정(Markov Decision Process, MDP)를 기반으로 부분 관측성(partial observability)을 고려한 decentralized partially observable Markov decision process(Dec-POMDP)로 모델링할 수 있으며, 튜플 $(n, S, O, A, \pi, P, r)$로 표현된다. 현재 상태 $s \in S$에서 군집 시스템에 포함된 n개의 개체가 선택한 공동 행동(joint action) $\boldsymbol{a} \in \boldsymbol{A} = A^n$의 결과 다음 상태 $s'$으로 변환하게 되며, 이러한 상태변화는 상태 변환 함수(state transition function) $P(s'|s, \boldsymbol{a})$으로 나타낼 수 있다. 각 개체가 인식하는 상태인 부분 관측 $o \in O(s, a)$를 토대로 통계적 결정함수인 정책 $\pi(a|o)$를 따라 다음 행동이 결정된다. 군집 시스템에서는 개별적인 이득보다는 공동의 목표 달성을 위해 공동행동 가치함수 $Q^{\boldsymbol{\pi}}(s_t, \boldsymbol{a}_t) = E[R_t|s_t, \boldsymbol{a}_t]$를 최대화하는 공동정책(joint policy) $\boldsymbol{\pi}$를 찾는 것이 목적이 된다. 여기서 $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$는 discounted return 값을 의미하며, 할인상수(discount factor) $\gamma \in [0,1)$를 어떻게 설정하느냐에 따라 최근 보상에 중점을 두는지 오래전의 보상에도 중점을 두는지가 결정된다. 군집의 모든 개체들은 일반적으로 같은 보상함수(reward function) $r(s, \boldsymbol{a})$을 공유한다.

### 2.2 Q-learning with QMIX[16]

다수 개체들이 취한 행동이 전체 시스템 입장에서 유리한 행동인지 판단하기 위해 공동행동 가치함수(joint action value function) $Q(s, \boldsymbol{a})$값을 계산하여 평가해야 하지만, 이를 직접적으로 사용하는 것은 차원의 저주 문제가 있기 때문에 DeepMind[17]의 연구에서는 value decomposition network(VDN)를 제안하여 이러한

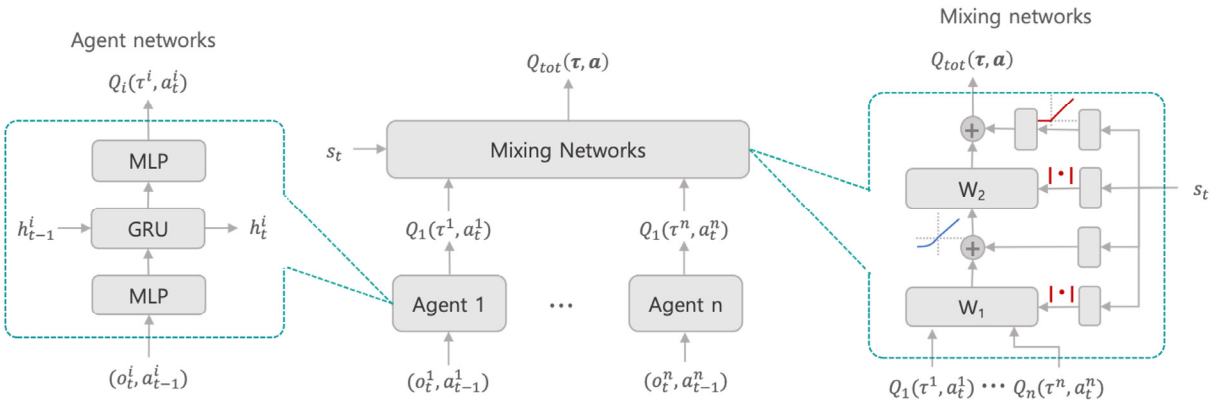


Fig. 1. Architecture of QMIX network[16], reproduced for better resolution

문제를 해결하고자 하였다. VDN은 각 개체의 행동을 결정하는 개별 네트워크(agent network)와 모든 개체의 행동이 주어졌을 때 전체 시스템에 유불리를 평가하는 통합 네트워크(mixing network)로 구성되어 있으며, 다수 개체들의 행동을 통합적으로 평가하기 위해 개별 네트워크의 가치함수(value function) $Q_i(s^i,a^i)$의 합 $Q_{tot}$을 구하는 방식(식 (1))을 사용하였다.

$$Q_{tot}(s,\boldsymbol{a}) = \sum_{i=1}^{n} Q_i(s^i,a^i), \quad (1)$$

$where\ s^i, a^i : state, action\ of\ agent\ i$

QMIX는 VDN과 같은 맥락에서 접근하되, 개별 네트워크와 통합 네트워크의 변화 방향은 같아야 한다는 단조성(monotonicity) 조건(식 (3))을 만족하도록 통합 네트워크를 구성하였고 개별 네트워크와 통합 네트워크에서 구하는 가치함수의 최대값을 다음 식 (2)와 같이 정의함으로써 두 네트워크 간의 일관성을 유지할 수 있도록 하였다.

$$\arg\max_{\boldsymbol{a}} Q_{tot}(s,\boldsymbol{a}) = \begin{pmatrix} \arg\max_{a_1} Q_1(s,a^1) \\ \vdots \\ \arg\max_{a_n} Q_n(s,a^n) \end{pmatrix} \quad (2)$$

$$\frac{\partial Q_{tot}}{\partial Q_i} \geq 0,\ \forall i \in N \quad (3)$$

QMIX에서 사용되는 deep Q-learning 기법은 θ를 매개변수로 가지는 심층 신경망(deep neural network)으로 가치함수 $Q(s,a;\theta)$를 모사하는 방식이다. Deep Q-networks(DQN)이라고도 하는 해당 모델은 시뮬레이션과의 상호작용(rollout)을 통해 재현 메모리(replay memory)에 저장한 전환 튜플(transition tuple) $(s,a,r,s')$을 샘플링하여 시간차(temporal difference, TD) 에러를 줄이도록 학습한다. DQN에 recurrent neural network (RNN)을 사용하면 단편의 전환(transition)을 사용하는 대신 전환의 모음인 경로(trajectory) τ를 입력 데이터로 학습함으로써 MDP의 순차적인 특성을 고려하도록 할 수 있다. QMIX나 VDN과 같은 접근법의 경우 개별 네트워크는 분산 환경에서 실행한 후, 통합 네트워크는 중앙화된 학습을 진행할 수 있다는 이점이 생긴다. QMIX 알고리즘에 사용되는 네트워크는 개별 네트워크에서 각 개체의 관측값에 대한 Q-value을 계산하여 행동 값을 결정하고, 통합 네트워크에서는 각 개체가 행동 선정에 사용한 Q-value 값과 전역 상태(global state) 값을 사용하여 전체 군집 시스템 레벨에서의 가치함수값인 $Q_{tot}$값을 계산하게 된다.

최종적으로 QMIX 네트워크는 다음 식 (4)와 같은 손실 함수(loss function)를 통해 end-to-end 방식으로 학습된다. 여기서 $b$는 학습에 사용되는 배치 크기(batch size), $\theta^-$는 일정한 주기로 업데이트되는 타겟 네트워크(target network)의 파라미터, 각 변수에 붙어 있는 ‘는 상태 전환 이후 값을 의미한다.

$$L(\theta) = \sum_{i=1}^{b} [y_i^{target} - Q \quad (4)$$

### 2.3 Transfer learning[15]

전이 학습은 이미 취득한 지식을 활용하여 한층 심화된 내용을 학습할 수 있는 기법으로, 기저가 되는 과업(task)을 학습한 후 좀 더 난이도가 있는 과업에 적용하는 방법은 크게 세 가지로 분류할 수 있다.: 1) Curriculum distillation, 2) buffer reuse, 3) model reload. 첫 번째 방식은 이전에 학습했던 과업의 정책과 새로 배울 과업의 정책 분포가 유사하다고 간주하는 방식, 두 번째 방식은 과업별로 모아둔 재현 메모리를 공유하여 현재 과업에서 사용하는 방식, 그리고 세 번째 방식은 이전 과업에 대해 학습한 모델을 다음 과업의 초기 모델로 사용하는 방식이다. Wang[15]의 연구에서 실험을 통해 군집의 수를 늘려가는 작업에서 model reload 방식이 가장 좋은 성능을 도출함을 보였다.

## 3. Methodologies

### 3.1 QMIX-TL : QMIX with transfer learning

본 연구에서는 군집을 이루는 개체의 수가 많아질 때 발생하는 확장성 문제를 해결하고자 다수 개체의 협력적인 행동에 대한 가치 평가가 가능하도록 모델을 구성하고 학습하여 학습 성능 및 효율을 높이도록 한 QMIX 알고리즘을 기반으로 하여, 적은 개체에 대한 학습에서 습득한 지식을 많은 개체를 제어하는 강화학습 모델에 전달하도록 하는 전이 학습을 도입하여 추가적인 학습 효율을 얻을 수 있도록 하였으며, 2.3 절에서 언급하였던 전이 학습 방법 중 model reload 방식을 적용하였다.

Algorithm 1. Pseudo-code of QMIX-TL

```
# Outer loop
For m in number of tasks do
    Initialize parameters
        θ_m ← θ_{m-1} if m > 1 else U(l,u)
    Initialize replay buffer D ← ∅
    # Inner loop
    For iteration in maximum iteration do
        Initialize episode buffer ε ← ∅
        # Collect samples
        For step in episode limit do
            Get observations o_t and states s_t
            Get action a_t
            Update environment s_{t+1}
            Calculate reward r_t
            Store transition ε ← ε ∪ (s_t, a_t, r_t)
        End for
        Compute returns R_t
        Update replay buffer D ← D ∪ ε
        # Training model
        For batch in train samples do
            Compute loss Eq.(4)
            Backpropagate
            Update parameters θ_m
        End for
        Update target network θ⁻
    End for
End for
```

학습 프로세스는 Algorithm 1에 기술된 바와 같이 진행되며, 내부 루프(inner loop)에서는 고정된 개체수에 대해 QMIX 모델을 이용한 다중 개체의 강화학습이 수행되고, 외부 루프(outer loop)에서는 지식을 전달하는 전이학습이 수행된다.

내부 루프의 QMIX 학습은 환경과의 상호작용(rollout)을 수행하여 샘플을 모아 재현 메모리에 축적한 후 일정 사이즈의 배치를 가지고 손실함수를 계산, 역전파를 통해 현 단계 모델의 파라미터 $\theta_m$의 업데이트 과정을 일정 수렴 조건이 만족될때까지 반복하게 된다.

외부 루프에서의 지식 전달을 위해 이전 단계에서 학습한 모델의 파라미터 $\theta_{m-1}$를 현재 단계에서 학습할 모델의 파라미터 $\theta_m$의 초기화에 사용하게 된다. 단, 이전 단계가 없는 경우 ($m$ = 1) 균등 분포(uniform distribution, $U$)를 따라 랜덤하게 초기화한다.

개체수가 증가하는 경우 모델 입력층(input layer)의 사이즈 차이가 발생하므로 확장된 차원에 파라미터를 복사하도록 하였다. 단, 개체들의 특성이 모두 동일한 동종(homogeneous) 군집이므로 이와 같은 방식으로 파라미터를 확장하는 데 문제가 없지만 개체들의 특성이 다른 이종(heterogeneous) 군집의 경우 단순 확장은 주의가 필요하다.

Wang[14]의 연구에서는 전이학습을 활용하는 과정에서 가변하는 개체수를 다루기 위해 그래프 신경망 네트워크(Graph Neural Network, GNN)을 사용하여 개별 객체의 상태를 임베딩한 후 취합(aggregation)하는 방식으로 군집의 상태를 표현하여 개별 네트워크 없이 통합 네트워크를 통해 Q-value를 평가하는 방식을 사용한 바가 있다. QMIX 기법이 이종 군집을 다룬 문제에서 기존의 기법 들에 비해 탁월한 성능을 보였기에 가변하는 이종 군집에 대한 임베딩 방식을 GNN 등의 기법을 활용한다면 본 연구가 가지고 있는 한계점을 해결할 수 있을 것이라 기대된다.

## 4. Experiments

### 4.1 Problem descriptions

3절에서 기술된 QMIX-TL 알고리즘을 통해 확장성 문제가 해결되는지 검증하기 위해 특정 도착 지점을 향해 움직이는 목표물을 따라 이동하는 군집 시스템의 제어 문제를 선정하였다. 움직이는 목표물이 아군인지 적군인지에 따라 방어나 공격의 기본이 되는 임무이므로, 난이도가 높은 임무에 학습 결과를 다시 활용할 수 있는 문제이다. 본 연구에서는 5개 개체가 움직이는 목표물로 이동하는 임무를 난이도가 낮은 task 1으로 설정하였고, task 2는 20개 개체가 움직이는 목표물로 이동하는 임무를 부여하되 5개 개체일 때와 달리 밀도가 높아 충돌하는 경우를 방지하기 위한 임무를 추가하였다.

### 4.2 Experiment setups

다수 개체 환경을 모의하기 위한 시뮬레이터로 픽셀 기반의 MAgent[18]를 사용하였으며, 100×100 크기의 2차원 맵에서 총 3가지 그룹 – *Agent, Target, Goal* – 을 포함하도록 구성하였다(Fig. 2). *Target*은 고정된 도

착 지점(*Goal*)을 향해 일정한 속도로 움직이고, 다수의 *Agent*는 움직이는 *Target*을 따라가도록 유도하기 위해 *Target*과의 거리가 가까울수록 높은 보상을 받도록 다음과 같은 Migrate reward를 설계하였다.

$$r_i^{migrate} = \begin{cases} 1 & \text{if } d_{ti} \le c_1 \\ -d_{ti} & else \end{cases}, \quad where\ d_{ti} = \| \boldsymbol{p}_t - \boldsymbol{p}_i \| \tag{5}$$

Crowded reward를 통해 개체수가 많아짐에 따라 발생할 수 있는 충돌을 방지하기 위해서 가장 가까운 개체와의 거리가 특정 반경 이하로 들어오는 경우 페널티를 부여하도록 하였다.

$$r_i^{crowded} = \begin{cases} -1 & \text{if } d_{ji} \le c_2 \\ 0 & else \end{cases}, \quad where\ d_{ji} = \min_j \| \boldsymbol{p}_j - \boldsymbol{p}_i \| \tag{6}$$

또한, 불필요한 행동을 취하지 않도록 한번 움직일 때마다 shortest path reward, $r_i^{shortest}$ = -0.01 만큼씩 페널티를 받도록 하였다. 최종적으로 $i$번째 객체가 받는 보상은 위의 세 가지 항을 더한 값이고(식 (7)), 전체 군집 시스템으로써 받는 보상은 각 객체들이 가지는 보상의 평균값을 가지도록 하였다(식 (8)).

$$r_i = r_i^{migrate} + r_i^{crowded} + r_i^{shortest} \tag{7}$$

$$r = \frac{1}{n}\sum_{i=1}^{n} r_i \tag{8}$$

Fig. 2에 도식화된 바와 같이 MAgent환경에서 개체가 취할 수 있는 행동은 9가지 종류의 이산화된 행동(discrete action)으로 정의하였다.

학습에 사용되는 입력값에는 개별 네트워크에서 각 객체가 움직일 방향을 결정하는 데 활용하는 관측값(observation) (Table 1)과 통합 네트워크에서 전체적인 상황 판단에 사용되는 전역 상태(global state) (Table 2)가 있다. 개별 개체의 관측값에는 일정 탐지 범위(view range)안에 탐지되는 이웃 개체들의 분포와 평균적인 이동 방향이 포함되고, 중요한 정보는 중앙에서 통보 (broadcast) 된다는 가정하에 목표물과 개별 객체의 상대적 위치 및 이동 방향이 추가되도록 하였다. 실제로 드론에는 주위의 근접 반경을 감지하는 카메라 등의 센서가 탑재되며, 중앙 통제 시스템에서 통신을 받을 수 있다는 현실적인 상황을 반영하도록 설정하였다.

QMIX의 통합 네트워크에서는 전체적인 상황에 대한 추가 변수를 활용하여 개별 네트워크 중에서 어떤 네트워크에 더 중점을 두어야 할지에 관한 파라미터를 학습하게 되는데, 이를 위한 전역 상태(global state) 변수에는 전체 객체들의 위치와 목표의 위치정보를 포함하고 있다.

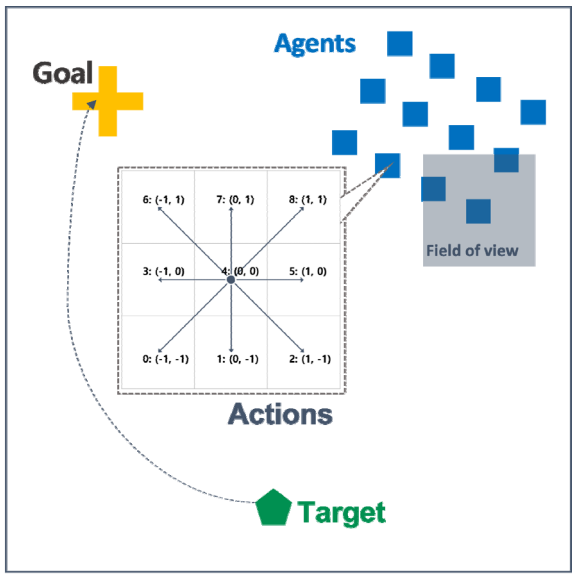


Fig. 2. Discrete action space on MAgent environment

Table 1. Components of observations

| Observations, $o_t$ | |
|---|---|
| Mean of relative positions (b.t.w. agent & neighbors) | $mean(\Delta_{ji} p_x)$ |
| | $mean(\Delta_{ji} p_y)$ |
| Mean of relative direction (b.t.w. agent & neighbors) | $mean(\Delta_{ji} a_x)$ |
| | $mean(\Delta_{ji} a_y)$ |
| Relative position (b.t.w. agent & target) | $\Delta_{ki} p_x$ |
| | $\Delta_{ki} p_y$ |
| Relative direction (b.t.w. agent & target) | $\Delta_{ki} a_x$ |
| | $\Delta_{ki} a_y$ |
| $i \in N,\ j \in \dot{N},\ k = 1,\ \Delta_{ji}(\bullet) = (\bullet)_j - (\bullet)_i$ | |

Table 2. Components of global states

| Global states, $s_t$ | |
|---|---|
| Absolute positions: agent | $p_{x,i}$, $p_{y,i}$ |
| Absolute positions: target | $p_{x,k}$, $p_{y,k}$ |
| Actions: agent | $a_{x,i}$, $a_{y,i}$ |
| Actions: target | $a_{x,k}$, $a_{y,k}$ |
| HPs: agent | $hp_i$ |
| HPs: target | $hp_k$ |
| $i \in N,\ j = N \setminus i,\ k = 1$ | |

학습 모델 파라미터 변수 최적화를 위해 root mean square propagation(RMSprop) 알고리즘을 사용하였으며, 학습률(learning rate)은 1e-3, 배치 크기(batch size)는 64, 각 에피소드는 최대 160 step을 가지고, return값을 계산하는데 사용하는 discount factor (γ)는 0.99으로 설정하였고, Q-value값을 근거로 행동을 선택하는 방식은 학습이 진행함에 따라 무작위 행동을 취하지 않도록 입실론(epsilon)값을 줄여나가도록 설정한 입실론 그리디(epsilon greedy) 방식을 사용하였다.

### 4.3 Experiment results

20개 개체에 대해 전이학습 없이 학습하는 DL (Direct Learning)과 전이학습을 통해 5개 개체에서 학습된 결과를 전달받은 TL(Transfer Learning)에 대해 학습을 통해 달성하는 보상값 return을 비교해 본 결과 (Fig. 3) TL의 경우 최종적으로 140정도의 return값을 획득하였지만 DL의 경우 10정도 수준에 머물러 처음부터 다수의 개체에 대해 학습을 진행하는 DL의 경우 학습에 난항을 겪고 있음을 확인할 수 있었다. 즉, 유사한 성격의 과업에 대해서는 적은 개체수로 학습 난이도를 낮춰 습득한 지식을 높은 난이도의 문제에 전달함으로써 개체의 수가 많아짐에 따라 발생하는 확장성 문제를 해결할 수 있음을 보여주는 실험결과이다.

DL방식으로는 TL방식 만큼의 학습 성능을 얻을 수 없었기 때문에 전이학습에 소요된 총 에피소드 수만큼을 DL에 소요된 시간으로 볼 수 있으며, 이에 따라 Table 3에 기술된 바와 같이 전이학습을 통해 학습 시간을 31 % 절감할 수 있다고 할 수 있다. Table 3에서 학습에 걸린 실 소요시간은 한 에피소드(160 steps)를 진행하는데 걸리는 평균 소요 시간과 학습이 진행

Table 3. Comparison of learning costs

| 구분 | 학습시간 계산 | 실소요시간 |
|---|---|---|
| TL | $t_{TL} = T_{small} \times t_{small} + T_{large} \times t_{large}$ | 367 hrs |
| DL | $t_{DL} = (T_{small} + T_{large}) \times t_{large}$ | 532 hrs |

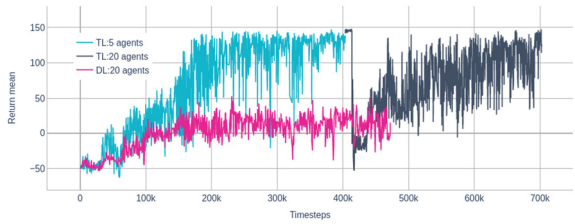


Fig. 3. Histories of return values

된 에피소드 수의 곱으로 구하였으며, Intel i7-10700를 사용한 계산환경에서 평균 소요 시간은 5개 개체 모델 $t_{small}$ = 1.25 seconds, 20개 개체 모델 $t_{large}$ = 2.74 seconds로 측정되었으며, 전이학습에서 5개 개체로 학습한 에피소드 수 $T_{small}$ = 400,000와 20개 개체로 학습한 에피소드 수 $T_{large}$ = 300,000를 기준으로 계산하였다.

실제로 학습된 정책을 따라 시뮬레이션을 수행해본 결과(Fig. 4) TL의 경우 목표물을 대부분 따라가는 반면 DL의 경우 목표물을 제대로 따라가지 못하고 있음을 확인할 수 있다.

### 4.4 Sensitivity Analysis

일반적으로 강화학습이 실용적인 문제에서 활용되지 못하고 있는 이유는 학습된 정책에 의한 의사결정에 관한 판단의 근거를 제시할 수 없다는 점이 크다고 알려져 있다[19,20]. 본 연구에서는 시뮬레이션 결과뿐 아니라 어떤 입력값을 근거로 이러한 행동을 내리게 되었는지 살펴보기 위해 각 객체 네트워크에서 관측값에 대한 Q-value 값의 민감도(sensitivity) $\partial Q_i / \partial O_i$를 구해보았으며, 목표물과의 거리가 멀어 목표물에 접근하는 것을 우선 사항으로 두고 행동을 취한 경우 목표와의 거리에 관련된 관측값(px_target, py_target)의 민감도가 크게 나타나는 반면(Fig. 5), 목표물과의 거리가 가까워 목표물과의 거리에 관련된 관측값(px_target, py_target)의 민감도는 앞선 경우(Fig. 5)에 비해 작아지고 다른 객체와의 거리가 가까울 경우 충돌을 피하

기 위한 목적이 우세해져 이웃 객체들과의 거리에 관련된 관측값(px_neighbors_mean, py_neighbors_mean)의 민감도가 이전(Fig. 5)에 비해 커진 것을 확인할 수 있었다(Fig. 6). 이 기능의 경우 pytorch에서 제공하는 자동 미분(automatic difference) 기능을 사용하였으며, 이미 학습된 파라미터를 활용하기 때문에 계산 부하가 크지 않아 다양한 RL 연구에 적용할 수 있을 것으로 기대된다.

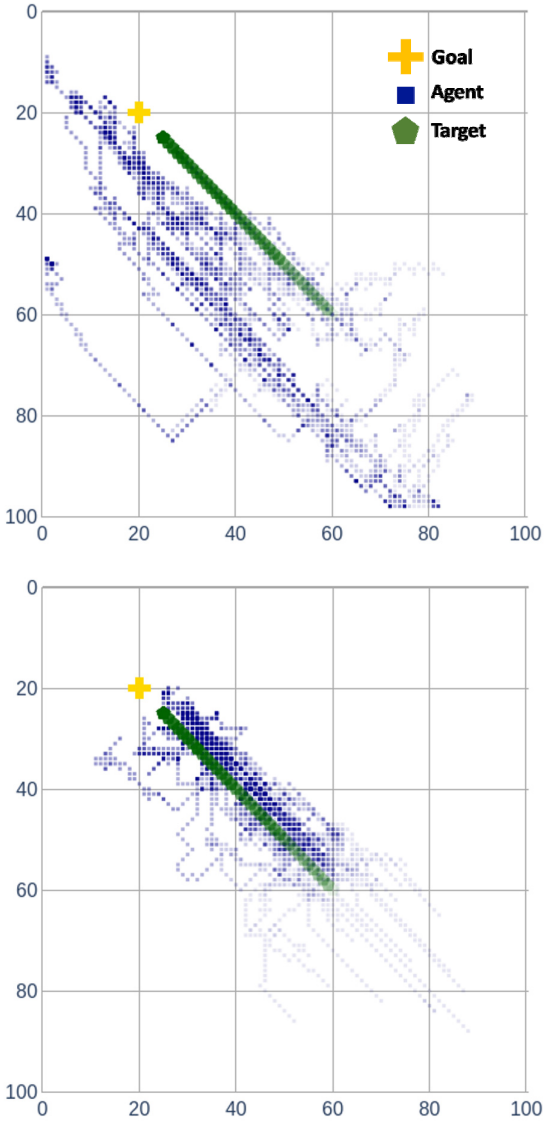


Fig. 4. Trajectories of simulation results
(upper: DL, lower: TL)

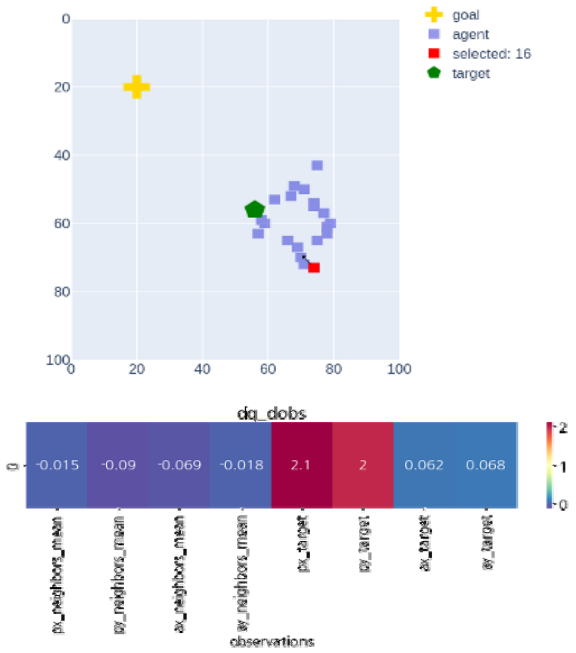


Fig. 5. Agents approaching target
(upper: status, lower: sensitivity)

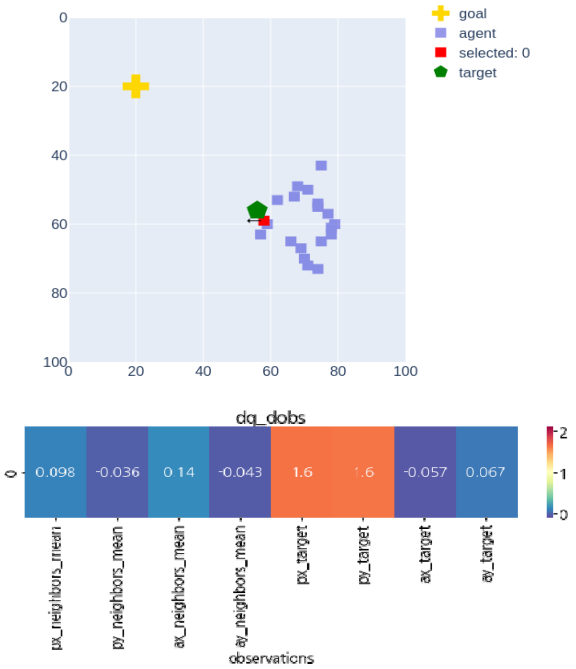


Fig. 6. Agents avoiding collision
(upper: status, lower: sensitivity)

## 5. Conclusions

본 연구를 통해 MARL에서 개체수 증가에 따라 학습 성능이 떨어지는 문제인 확장성 문제가 실제로 발생한다는 것을 확인하였고, 이를 위한 해결책으로 적은 개체에 대해 학습한 결과를 많은 개체수를 위한 모델에 전달하는 전이학습을 통해 효율적으로 학습 목표를 달성할 수 있음을 실험을 통해 확인하였다. 또한 민감도 분석을 통해 학습된 강화학습의 결과가 어떤 관측값을 근거로 이러한 행동을 결정하였는지 살펴보았다. 본 연구에서 적용한 전이학습은 유사한 임무에 대해서 개체수를 증가시키는 경우에는 효율적으로 학습을 수행하는 데 유용함을 보였으나, 특성이 다른 임무나 이종 군집에 대한 효과는 추후 연구에서 확인할 예정이다.

## 후 기

이 논문은 2023년 정부(방위사업청)의 재원으로 국방과학연구소의 지원을 받아 수행된 연구임.
(UG190055RD)

## References

[1] https://www.newspim.com/news/view/20230106000810
[2] https://www.bbc.com/korean/international-49705340
[3] https://www.darpa.mil/work-with-us/offensive-swarm-enabled-tactics
[4] Lowe, R. et al., “Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments,” Proceedings of 31st Conference on Neural Information Processing Systems(NIPS 2017), Long Beach, CA, USA, 2017, arXiv preprint arXiv:1706.02275
[5] Vinyals, O. et al., “Grandmaster Level in StarCraft II using Multi-Agent Reinforcement Learning,” Nature, Vol. 575, pp. 350-369, 2019.
[6] Hasan, Y. W. et al., “Defensive Escort Teams for Navigation in Crowds via Multi-Agent Deep Reinforcement Learning,” Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), Las Vegas, USA, 2020.
[7] Rahman, A. et al., “AdverSAR: Adversarial Search and Rescue via Multi-Agent Reinforcement Learning,” 2022. arXiv preprint arXiv: 2212.10064v1
[8] Wang, J. et al., “Pattern RL: Multi-robot Cooperative Pattern Formation via Deep Reinforcement Learning,” Proceeding of 18th IEEE International Conference on Machine Learning and Applications, 2019.
[9] Reynolds, C. W., “Flocks, Herds, and Schools: A Distributed Behavioral Model,” Computer Graphics, Vol. 21(4), pp. 25-34, 1987.
[10] Liu, Y. et al., “A Survey of Formation Control and Motion Planning of Multiple Unmanned Vehicles,” Robotica, Vol. 36(7), pp. 1019-1047, 2018.
[11] Hüttenrauch, M. et al., “Deep Reinforcement Learning for Swarm Systems,” Journal of Machine Learning Research, Vol. 20, pp. 1-31, 2019.
[12] Zhou, J., “Adversarial Swarm Defense with Decentralized Swarms,” M.S. Thesis, Univ. of California, Berkeley, 2021.
[13] Zhang, K. et al., “Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms,” Handbook of Reinforcement Learning and Control. Studies in Systems, Decision and Control, Vol. 325, Springer, pp. 321-384, 2021. arXiv preprint arXiv:1911.10635
[14] Liu, X. et al., “Attentive Relational State Representation in Decentralized Multiagent Reinforcement Learning,” IEEE Transactions on Cybernetics, Vol. 52(1), pp. 252-264, 2022.
[15] Wang, W. et al., “From Few to More: Large-scale Dynamic Multiagent Curriculum Learning,” Proceedings of AAAI Conference on Artificial Intelligence(AAAI), 2020. arXiv preprint arXiv: 1909.02790
[16] Rashid T. et al., “QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning,” Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. arXiv preprint arXiv: 1803.11485
[17] Sunehag, P. et al., “Value-Decomposition Networks for Cooperative Multi-Agent Learning,” 2017. arXiv preprint arXiv:1706.05296
[18] Zheng, L. et al., “MAgent: A Many-Agent

Reinforcement Learning Platform for Artificial Collective Intelligence," NIPS 2017 & AAAI 2018 Demo, 2017. arXiv preprint arXiv:1712.00600

[19] Lu, M. et al., "Is Deep Reinforcement Learning Ready for Practical Applications in Healthcare? A Sensitivity Analysis of Duel-DDQN for Hemodynamic Management in Sepsis Patients," AMIA Annual Symposium Proceedings, pp. 773-782, 2020. arXiv preprint arXiv: 2005.04301

[20] Boggess, K. et al., "Toward Policy Explanations for Multi-Agent Reinforcement Learning," 2022. arXiv preprint arXiv:2204.12568v1