Journal of Next-generation Convergence Information Services Technology Vol.15, Suppl.1, November (2026) pp.89-102
ISSN 2384-101X(p) 2672-1163(e) *http://dx.doi.org/10.29056/jncist.2026.11.08*

2대2 피구 게임에서 자기대전 기반 다중 에이전트 강화학습의 적용과 에이전트 행동 분석

Application of Self-Play Multi-Agent Reinforcement Learning to a Two-on-Two Dodgeball Game and Analysis of Agent Behavior

박성은[1], 최재용[2]*

Sung-Eun Park[1], Andrew Jaeyong Choi[2]*

요 약

본 연구는 모방학습 없이 자기대전으로 학습한 에이전트가 소규모 팀 경쟁 게임에서 보이는 행동을 분석하고, 다중 에이전트 강화학습의 게임 AI 개발 활용 가능성과 한계 및 개선 방향을 논의한다. Unity ML-Agents로 2대2 피구 환경을 구현하고 관찰 공간, 행동 공간, 개인·팀 보상을 설계한 뒤 다중 에이전트 강화학습 알고리즘인 MA-POCA와 자기대전을 적용하여 약 1억 스텝을 학습하였다. 학습된 정책을 고정하여 100경기를 진행하고, 위치 점유와 투척 및 팀원 간 패스 기록을 분석하였다. 그 결과 공을 보유한 팀에 따른 위치 점유 차이, 전방 팀원에게 전달하는 패스와 후속 투척, 상대 무적 상태에 따른 투척 빈도 차이가 관찰되었다. 이는 플레이 시연과 세부 행동 규칙 없이도 경기 상황에 반응하고 팀원과 행동을 연계하는 AI를 구성할 수 있음을 보여준다. 이러한 결과를 바탕으로 소규모 게임 개발에서 행동 설계 부담과 플레이 데이터 의존도를 줄일 가능성을 살펴보고, 보상 설계, 학습 비용, 정책의 적응성에 관한 과제와 개선 방향을 제시한다.

핵심어 : 다중 에이전트 강화학습, 자기대전, MA-POCA, 게임 인공지능, 행동 분석

Abstract

This study analyzes the behavior of agents trained through self-play without imitation learning in a small-scale competitive team game and discusses the potential, limitations, and improvement directions of multi-agent reinforcement learning for game AI development. A two-on-two dodgeball environment was implemented using Unity ML-Agents. After designing the observation space, action space, and individual and team rewards, the agents were trained for approximately 100 million steps using MA-POCA, a multi-agent reinforcement learning algorithm, combined with self-play. The trained policy was fixed and

1 Department of Artificial Intelligence, Gachon University, Seongnam, Korea [Graduate Student]
e-mail: bluecar21@gachon.ac.kr
2 Department of Artificial Intelligence, Gachon University, Seongnam, Korea [Professor]
e-mail: andrewjchoi@gachon.ac.kr (Corresponding author)

*본 논문은 2026년 차세대컨버전스정보서비스학회 하계학술대회에서 발표된 논문을 수정·보완하여 확장한 논문이며 과학기술정보통신부 및 정보통신기획평가원의 생성AI선도인재양성사업 연구 결과로 수행되었음 (IITP-2026-RS-2026-25545512).

Received(October 1, 2026), Review Result(1st: October 19, 2026), Accepted(November 12, 2026), Published(November 30, 2026)

© 2026 The Authors. Published by NCISS.
This is an open access article licensed under the Creative Commons Attribution-NonCommercial 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc/4.0/.

evaluated over 100 matches by analyzing spatial occupancy, throws, and teammate-pass records. The results revealed differences in spatial occupancy depending on which team possessed the ball, forward passes and follow-up throws after pass reception, and differences in throwing frequency depending on opponents' invincibility. These findings demonstrate that game AI can respond to game situations and coordinate actions with teammates without gameplay demonstrations or explicitly prescribed rules for individual behaviors. Based on these results, this study examines the potential to reduce behavioral scripting and dependence on gameplay data in small-scale game development and presents challenges and improvement directions concerning reward design, training costs, and policy adaptability.

Keyword : Multi-Agent Reinforcement Learning, Self-Play, MA-POCA, Game AI, Behavioral Analysis

1. 서론

팀 경쟁 게임의 AI는 상대의 움직임에 대응하면서 이동, 공격, 방어와 팀원 간 연계를 선택해야 한다. 규칙 기반 개발에서는 이러한 행동을 개발자가 직접 정의하고 상황별 전환 조건을 작성한다. 그러나 상대와 팀원의 상태가 함께 변하는 환경에서는 고려할 상황과 예외 처리가 늘어난다. 다중 에이전트 강화학습(Multi-Agent Reinforcement Learning, MARL)은 여러 에이전트가 상호작용하며 보상을 통해 행동 정책을 학습하도록 하는 접근으로, 협력과 경쟁이 결합된 게임 AI를 구성하는 데 활용된다 [1-3].

본 연구는 플레이 시연 데이터가 없는 조건에서 소규모 팀 게임 AI를 학습시키고, 학습된 행동을 분석하는 데 초점을 둔다. 이를 위해 두 팀이 공 한 개를 공유하는 2대2 피구 환경을 구성하였다. 공을 보유한 팀에 따라 공격과 방어 상황이 바뀌며, 공 보유자는 직접 투척하거나 팀원에게 공을 전달할 수 있다. 팀원 한 명이 탈락한 뒤에도 경기가 계속되고, 일정 시간 피격을 막는 무적 아이템이 공격 시점과 위치 선택에 영향을 줄 수 있다. 이러한 규칙은 적은 수의 에이전트로도 경쟁과 팀원 간 상호작용을 함께 관찰할 수 있게 한다.

에이전트는 모방학습 없이 MA-POCA와 자기대전으로 학습하였다. 이후 정책을 고정하고 100경기의 기록을 수집하여 경기 상황별 행동 특성과 팀원 간 협력적 상호작용을 분석하였다. 구체적으로 공 보유 조건별 위치 점유, 패스의 방향과 후속 행동, 상대 무적 상태별 투척 빈도를 살펴보았다. 이를 통해 본 실험과 같은 소규모 팀 경쟁 게임에서 시연 데이터 없이 게임 AI를 학습시키는 가능성을 검토하고, 이러한 개발 방식의 장점과 한계, 개선 방향 및 활용 전망을 논의한다.

2. 연구 배경

2.1 협력과 경쟁이 결합된 학습

팀 경쟁 환경에서는 개인에게 유리한 행동과 팀의 목표에 도움이 되는 행동이 달라질 수 있다. 공 보유자가 직접 공격하는 대신 전방 팀원에게 공을 전달하는 행동이 그 예다. 따라서 개별 에이

전트의 행동과 팀 전체의 결과를 연결하는 보상 설계가 중요하다. 학습 시 여러 에이전트의 정보를 활용하고 실행 시 각 에이전트가 자신의 관측으로 행동하는 중앙집중 학습·분산 실행 방식은 이러한 문제를 다루는 대표적인 접근이다 [1][4][5].

QMIX와 MAPPO는 학습 시 여러 에이전트의 정보를 활용하고 실행 시에는 각 에이전트가 자신의 관측으로 행동하는 중앙집중 학습·분산 실행 방식의 협력형 MARL 방법이다. QMIX는 에이전트별 행동 가치를 결합하여 팀의 공동 행동 가치를 학습하며 [4], MAPPO는 PPO 기반 정책 학습에 중앙집중 가치 평가를 활용한다 [5]. 두 방법은 개별 에이전트의 행동을 팀 성과와 연결한다는 공통점을 가지지만, 이를 구현하는 학습 구조는 서로 다르다.

본 연구에서 사용하는 MA-POCA 역시 중앙집중 학습·분산 실행에 기반한 팀 학습 방법으로, 특히 에이전트가 에피소드 도중 제거될 수 있는 상황을 다룬다. 제거 이전의 행동을 이후에 발생한 팀 성과와 연결하여 학습할 수 있다는 특징이 있다 [6]. 본 연구의 피구 환경에서는 한 팀원이 탈락한 뒤에도 남은 팀원이 경기를 계속하므로, 이러한 환경 특성을 고려하여 MA-POCA를 적용하였다.

2.2 관련 연구

Baker 등은 숨바꼭질 환경에서 다중 에이전트의 경쟁을 통해 도구 활용과 협력이 결합된 전략이 발달하는 과정을 제시하였다 [3]. 에이전트는 물체로 은신 공간을 만들거나 상대의 전략에 대응하는 행동을 학습하였다. Jaderberg 등은 Quake III Arena의 깃발 탈취 환경에서 에이전트 집단을 학습시키고 인간 참가자를 포함한 대전 평가를 수행하여, 3차원 팀 경쟁 게임에서 강화학습의 활용 가능성을 보여주었다 [7]. 이들 연구는 게임 목표와 다른 에이전트와의 상호작용을 통해 상황에 따른 행동과 팀 단위 전략이 학습될 수 있음을 보여준다.

Unity ML-Agents는 게임 엔진의 물리 환경과 센서를 강화학습에 연결하는 도구다 [8]. Unity의 공식 DodgeBall 사례는 여러 공을 사용하는 팀 대항 환경에 MA-POCA와 자기대전을 적용하였다 [9]. 에이전트는 최대 네 개의 공을 보유할 수 있으며, 제거전과 깃발 탈취 모드에서 이동·투척·돌진을 수행한다. 이에 비해 본 연구는 두 팀이 공 한 개를 공유하고 각 진영에서 이동하는 2대2 환경을 구성하였다. 이러한 구성에서는 공 보유 주체의 변화, 팀원에게 공을 전달하는 선택, 일시적인 무적 상태가 공간 활용과 투척 시점에 어떻게 연결되는지를 집중적으로 분석할 수 있다.

TiZero는 시연 데이터 없이 11대11 축구를 학습하기 위해 적응형 커리큘럼과 자기대전 전략 등을 제안하고 기존 시스템과 성능을 비교하였다 [10]. 본 연구는 소규모 환경에 기존 학습 알고리즘을 적용한 뒤, 공 보유와 무적 상태에 따른 행동 및 패스 연계를 해석하는 데 초점을 둔다.

모방학습을 활용한 접근으로는 전문가 시연에서 정책을 학습하는 적대적 모방학습 방법인 GAIL이 있다 [11]. Cavadas 등은 플레이어 기록과 GAIL 기반 모방학습으로 인간과 유사한 NPC를 학습

시키고, Unity DodgeBall 환경에서 사용자 평가를 수행하였다 [12]. 본 연구는 시연 없이 보상과 자기대전으로 학습한 에이전트의 상황별 행동과 팀원 간 상호작용을 경기 기록으로 분석한다.

본 연구의 기여는 공 하나와 팀별 무적 아이템이 있는 소규모 팀 경쟁 환경에서 관찰·행동 공간과 개인·팀 보상을 설계하고, 고정 정책의 100경기를 조건별로 분석한 데 있다. 공 보유 팀별 공간 점유, 성공 패스의 방향과 후속 투척, 상대 무적 상태별 공 보유 시간당 투척 빈도를 바탕으로 소규모 게임 AI의 학습 설계와 평가 과제를 논의한다.

2.3 연구 질문

본 연구는 모방학습 없는 자기대전 기반 MARL의 행동 분석과 소규모 게임 AI 개발에서의 활용을 살펴보기 위해 다음 네 가지 연구 질문을 설정하였다.

RQ1. 모방학습 없이 자기대전으로 학습된 MARL 에이전트는 소규모 팀 경쟁 환경에서 어떤 행동 특성을 나타내는가?

RQ2. 학습된 에이전트 사이에서 어떠한 팀원 간 상호작용 및 협력 행동이 나타나는가?

RQ3. 모방학습 없이 자기대전 기반 MARL을 소규모 팀 경쟁 게임 AI 개발에 적용할 때의 장점과 한계는 무엇인가?

RQ4. 소규모 게임 AI 개발에 MARL을 적용하기 위해 어떤 개선 방향을 고려할 수 있는가?

RQ1과 RQ2는 경기 로그의 정량적 분석으로 답한다. RQ3과 RQ4는 관찰·행동·보상의 구현 내용과 행동 분석 결과를 근거로 장점과 한계 및 개선 방향을 도출하는 논의형 질문이다. 개발 비용의 절감이나 개선 방법의 효과는 후속 비교 평가 대상으로 구분한다.

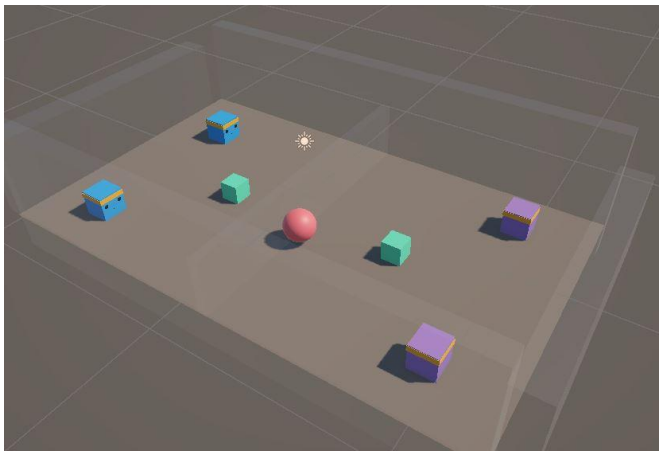
3. 게임 환경

3.1 경기장 구성

Unity 게임 엔진과 Unity ML-Agents 4.0.0을 사용하여 2대2 피구 환경을 구현하였다. Unity의 물리 엔진으로 에이전트와 공의 이동 및 충돌을 처리하고, ML-Agents를 통해 관찰 정보와 행동을 학습 시스템에 연결하였다.

[그림 1]의 경기장은 중앙선을 기준으로 두 진영으로 나뉜 3차원 직사각형 공간이다. 블루 팀과 퍼플 팀에 각각 에이전트 2명을 배치하였다. 외곽의 보이지 않는 벽은 공과 에이전트의 이탈을 막으며, 중앙의 경계는 공만 통과할 수 있도록 구성하였다. 경기에는 공 1개와 팀별 무적 아이템 1개가 사용된다. 공은 경기 시작 시 중앙 부근의 무작위 위치에, 아이템은 각 팀 진영의 무작위 위치

에 배치된다.



[그림 1] 2대2 피구 경기장과 주요 구성 요소

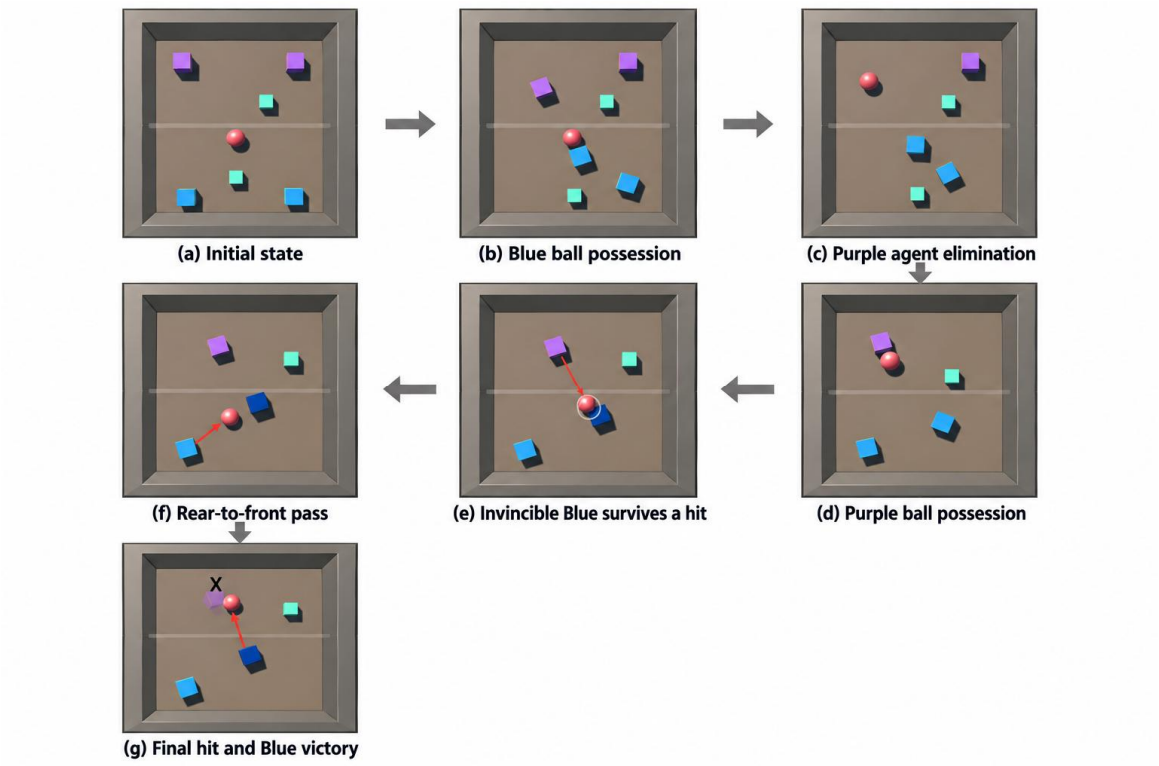
[Fig. 1] Two-on-two dodgeball arena and its main components

3.2 경기 규칙

에이전트는 공을 획득하여 보유하고, 자신이 바라보는 방향으로 투척할 수 있다. 투척 직후 공은 1초 동안 상대를 탈락시킬 수 있는 활성 상태가 된다. 활성 상태의 공이 무적이 아닌 상대에게 닿으면 해당 에이전트가 탈락하며, 팀원에게 전달된 공은 팀원이 받아 다시 보유할 수 있다. 한 팀의 두 에이전트가 모두 탈락하면 상대 팀의 승리로 경기가 종료된다.

자기 팀 아이템을 획득한 에이전트는 5초 동안 무적 상태가 되어 상대 공에 맞아도 탈락하지 않는다. 획득된 아이템은 5초 후 해당 진영의 무작위 위치에 재생성된다. 평가 경기의 제한은 3,000 물리 스텝으로, 물리 스텝 간격 0.02초를 기준으로 약 60초에 해당한다. 제한에 도달하면 시간 초과 무승부로 종료한다.

[그림 2]는 실제 게임 환경의 시각적 요소와 주요 경기 상황을 간소화하여 재구성한 경기 진행 예시이다. 경기 시작(a) 이후 블루 팀이 공을 획득하고(b), 공격으로 퍼플 팀 에이전트 한 명을 탈락시킨다(c). 남은 퍼플 팀 에이전트가 공을 획득하여 공격하지만(d), 무적 아이템을 획득한 블루 팀 에이전트는 피격에도 탈락하지 않는다(e). 무적 상태는 에이전트의 색상을 진하게 표시하여 구분하였다. 이후 후방 블루 팀 에이전트가 전방 팀원에게 패스하고(f), 공을 받은 팀원이 마지막 퍼플 팀 에이전트를 맞혀 블루 팀이 승리한다(g).



[그림 2] 피구 게임의 주요 규칙과 경기 진행을 간소화한 예시 (AI를 활용하여 생성한 예시 이미지)

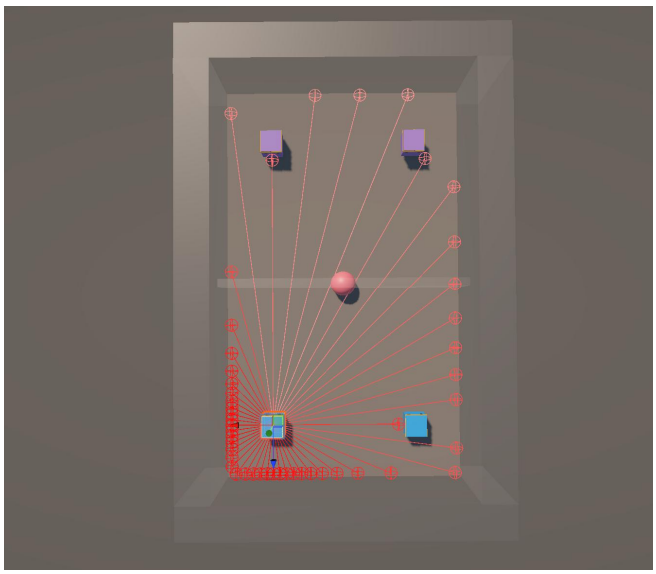
[Fig. 2] Simplified illustration of key rules and match progression in the dodgeball game (AI-generated illustrative image)

4. 에이전트 설계 및 학습 방법

4.1 관찰 공간과 행동 공간

관찰 공간은 27개의 벡터 관찰값과 레이 센서 정보로 구성하였다. 벡터 관찰에는 자신의 상태, 공의 상대 위치·속도, 팀원의 상태 및 자기 팀 아이템 정보가 포함된다. 공의 상대 위치·속도와 아이템의 상대 위치는 에이전트 기준의 로컬 좌표로 제공하고, 일부 관찰값은 고정 상수로 나누어 입력 크기를 조정하였다.

레이 센서는 [그림 3]과 같이 벽과 양 팀 에이전트를 관측하며, 팀원의 위치도 레이 센서로 확인한다. 좌우 각 24개 방향과 정면을 탐지하도록 구성하고, 최대 각도는 180도, 구형 탐지 반경은 0.3, 관측 스택은 2로 설정하였다. 관찰 정보의 구성은 [표 1]과 같다.



[그림 3] 에이전트의 레이 센서 관측 시각화

[Fig. 3] Visualization of an agent's ray sensor observations

[표 1] 벡터 관찰과 레이 센서의 구성

[Table 1] Vector observations and ray sensor inputs

구분	주요 정보
자기 상태	팀, 공 보유, 무적 여부 및 남은 시간, 이동 속도, 전방 방향, 중앙선까지의 거리
공 상태	상대 위치 및 속도, 자기 진영 여부, 활성 여부, 마지막 투척자와의 팀 관계
팀 상태	양 팀의 정규화된 탈락 수, 팀원의 생존 여부, 공 보유 여부, 무적 여부
아이템	자기 팀 아이템의 존재 여부와 상대 위치
레이 센서	벽과 양 팀 에이전트의 거리 및 탐지 태그 정보

행동 공간은 [표 2]와 같이 네 가지 이산 행동 분기로 구성하며, 각 의사결정 시 분기별로 하나의 행동을 선택한다. 의사결정 주기는 5 물리 스텝이고, 공을 보유하지 않은 경우 투척 행동을 마스킹한다. 공격과 패스에는 동일한 투척 행동을 사용한다.

[표 2] 에이전트의 행동 공간 구성

[Table 2] Agent action space configuration

행동 분기	선택 가능한 행동	선택지 수
전후 이동	정지, 전진, 후진	3
좌우 이동	정지, 좌측 이동, 우측 이동	3
회전	회전 없음, 좌회전, 우회전	3
투척	투척하지 않음, 투척	2

4.2 보상 함수

상대 적중과 피격에는 개인 보상과 팀 보상을 함께 부여하고, 경기 승패에는 팀 보상을 적용하였다. 또한 접근하는 상대 공을 위협으로 감지한 뒤 일정 시간 생존하고, 공이 지나가거나 활성 상태가 종료된 경우에는 개인 회피 보상을 부여하였다. 같은 위협에 대한 중복 보상은 제한하였다. 보상값은 [표 3]과 같다.

[표 3] 개인 보상과 팀 보상의 구성
[Table 3] Individual and team reward design

사건	개인 보상	팀 보상
경기 승리 / 패배	없음	+1 / -1
상대 적중	실제 투척자 +0.15	투척 팀 +0.05
상대 공에 피격	피격자 -0.15	피격 팀 -0.05
위협 공 회피	해당 에이전트 +0.02	없음
성공한 패스	직접 보상 없음	직접 보상 없음

패스에는 직접 보상을 부여하지 않았으며, 학습 후 나타난 패스와 후속 행동을 팀원 간 연계 행동으로 분석하였다.

4.3 학습 설정과 자기대전

학습에는 MA-POCA와 자기대전을 적용하였다. 팀별 그룹을 구성하여 개인 보상과 팀 보상을 함께 사용하고, 현재 정책과 과거에 저장된 정책을 상대하며 학습하였다. 모방학습과 플레이 시연 데이터는 사용하지 않았으며, 환경에서 제공하는 보상만으로 정책을 학습했다. 주요 학습 설정은 [표 4]와 같다. 총 학습량은 약 1억 trainer step으로, 학습 에이전트들이 수집한 경험을 누적한 값이다 [13].

경기장 5개를 병렬로 실행하여 동일한 정책의 학습에 필요한 경험을 수집하였다. 학습 장비는 Intel Core i5-13600KF CPU, 32 GB RAM, NVIDIA GeForce RTX 3060 Ti GPU이며, 학습에는 약 80시간이 소요되었다.

[표 4] MA-POCA 학습과 자기대전의 주요 설정
[Table 4] Main settings for MA-POCA training and self-play

설정 항목	값
학습기 / 최대 trainer step	POCA / 100,000,000
은닉층 수 / 층별 유닛 수	2 / 256

ML-Agents 자동 관측 정규화	사용하지 않음
배치 크기 / 버퍼 크기	2,048 / 20,480
학습률 / 학습률 스케줄	0.0003 / constant
엔트로피 계수 beta / 클리핑 epsilon	0.005 / 0.2
GAE lambda / epoch 수	0.95 / 5
외재적 보상 gamma / strength	0.99 / 1.0
time_horizon	128
자기대전 save_steps / team_change	100,000 / 500,000
상대 교체 swap_steps	100,000 ghost steps
과거 정책 window / 최신 정책 상대 비율	10 / 0.5
체크포인트 간격 / 최대 보관 수	2,000,000 / 50

무적 효과의 지속 시간은 학습 진행에 따라 0초, 2.5초, 5초로 증가시키는 커리큘럼으로 설정하였다. 단계 전환에는 학습 스텝을 최대 학습 스텝으로 나눈 진행률을 사용하였다 [14]. 최대 학습량인 1억 스텝을 기준으로, 200만 스텝과 500만 스텝을 각각 다음 단계로의 전환 기준으로 설정하였다. 학습된 정책의 평가는 무적 지속 시간이 5초인 환경에서 수행하였다.

5. 평가 결과

최종 학습 정책 한 개를 양 팀의 네 에이전트에 동일하게 적용하고, 추가 학습 없이 100경기를 평가하였다. 행동은 확률적으로 선택하고 경기마다 초기화 시드를 변경하였다. 경기별 승패와 투척·패스 등의 사건을 기록했으며, 에이전트의 상태는 약 0.1초 간격으로 수집하였다.

5.1 경기와 행동 개요

전체 경기 결과와 주요 행동 횟수는 [표 5]와 같다. 전체 투척 수에는 공격과 패스가 모두 포함되며, 성공 패스는 팀원이 실제로 공을 받은 사건을 의미한다.

[표 5] 고정 정책의 100경기 평가 결과
[Table 5] Overview of 100 matches with a fixed policy

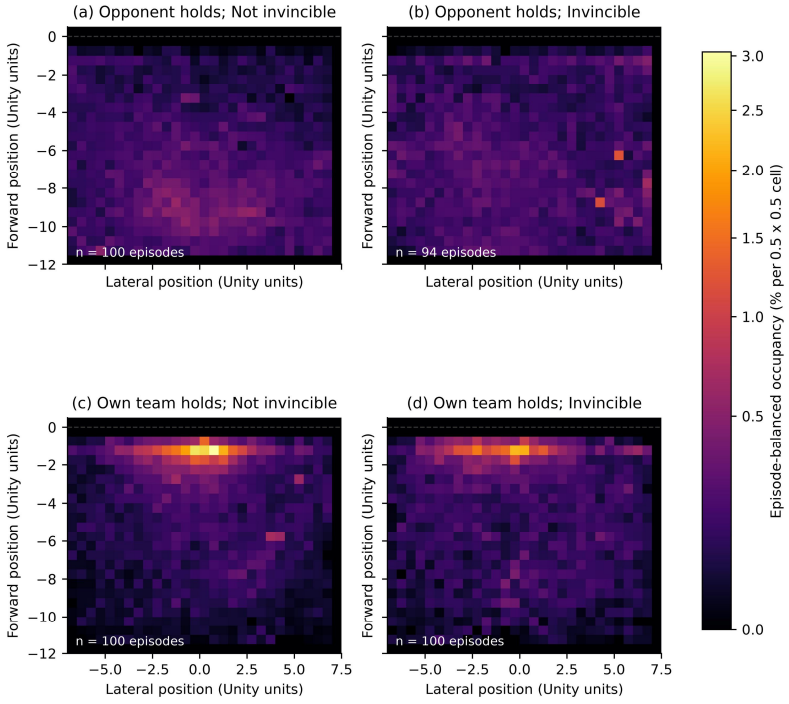
평가 지표	결과
완료 경기	100경기
블루 팀 승리 / 퍼플 팀 승리 / 시간 초과 무승부	39 / 41 / 20경기
평균 경기 시간	33.94초
전체 투척 수(패스 포함)	887회
상대 적중 사건	256회
성공적으로 수신된 팀원 패스	133회

성공 패스가 발생한 경기	64경기
아이템 획득	627회

100경기 중 80경기는 한 팀의 전원 탈락으로, 20경기는 시간 초과로 종료되었다. 팀원 간 성공 패스는 전체 경기의 64%에서 나타나, 공 전달이 여러 경기에서 반복적으로 관찰되었다.

5.2 공 보유와 무적 상태에 따른 공간 활용

양 팀의 위치를 공격 방향이 같도록 정렬하고 중앙선을 전방 좌표 0으로 두었다. 자기 진영의 좌표는 음수이며, 0에 가까울수록 중앙선에 가깝다. Unity 좌표 단위는 실제 미터와 구분한다. 살아 있는 에이전트만 대상으로 공 보유 팀과 관측 에이전트 자신의 무적 여부를 조합하여 네 조건으로 나누고, 공을 아무도 보유하지 않은 시점은 제외하였다. 자기 팀 공 보유 조건에는 공을 직접 보유한 에이전트와 공을 보유하지 않은 팀원이 모두 포함된다. 각 위치의 체류 시간은 해당 기록 시점부터 다음 기록 시점까지의 시간으로 계산하였다.



[그림 4] 공 보유 팀과 에이전트의 무적 상태에 따른 위치 점유

[Fig. 4] Spatial occupancy by ball possession and agent invincibility

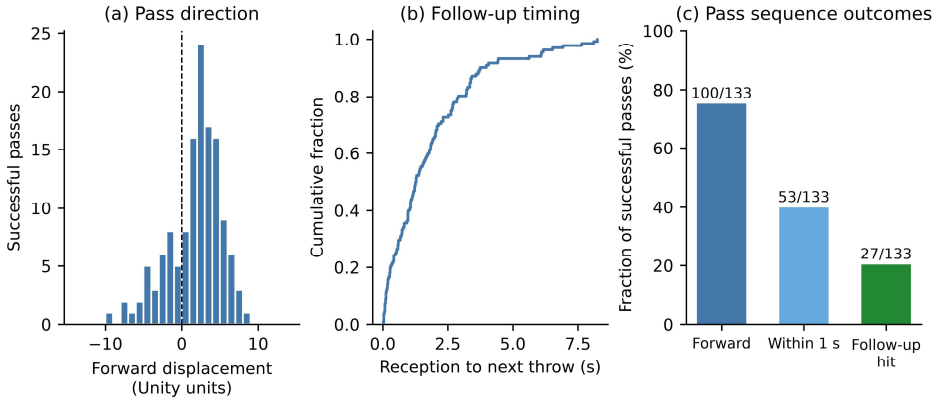
[그림 4]는 0.5 × 0.5단위 격자별 체류 시간을 경기별·조건별로 정규화하고 경기에 동일한 가중

치를 주어 평균한 위치 점유 비율이다. 색은 이동 속도나 이동 경로가 아니라 해당 위치에 머문 비율을 나타낸다. 평활화는 적용하지 않았으며, 네 패널에 동일한 범위와 감마 0.55의 비선형 색상 정규화를 사용하여 낮은 점유량도 구분되도록 하였다. 상대 팀 보유·무적 조건은 유효 관측이 있는 94경기를 사용했고, 나머지 조건은 각각 100경기를 사용하였다.

상대 팀이 공을 보유하고 관측 에이전트가 비무적일 때에는 자기 진영 후방에 위치 점유가 넓게 분포하였다. 반면 자기 팀이 공을 보유할 때에는 중앙선 가까이에 점유가 집중되었다. 상대 팀이 공을 보유한 상황에서는 무적 상태에서 비무적 상태보다 상대적으로 전방에 분포하는 양상이 나타났으나, 자기 팀이 공을 보유한 상황에서는 이러한 양상이 뚜렷하지 않았다. 이는 공 보유 상황과 무적 여부에 따라 에이전트가 점유하는 공간이 달라지는 양상을 보여준다.

5.3 팀원 패스와 후속 행동

성공 패스는 투척한 공을 같은 팀의 다른 에이전트가 실제로 받은 사건으로 정의하였다. 송신자의 투척 위치보다 수신자의 수신 위치가 중앙선에 가까우면 전방 패스로 분류하였다. 100경기에서 성공 패스는 133회였고, 이 중 100회(75.2%)가 전방 패스였다. [그림 5](a)는 이러한 패스 방향의 분포를 나타낸다.



[그림 5] 성공 패스의 방향과 후속 투척 및 적중 연계

[Fig. 5] Successful pass directions and subsequent throw and hit sequences

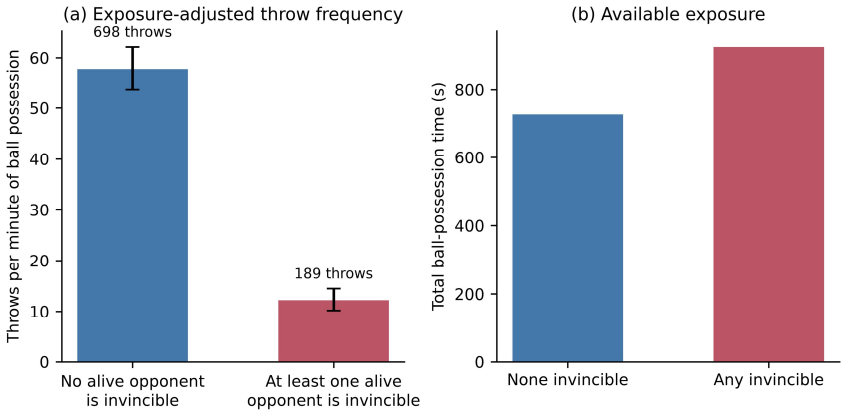
패스 수신 후 수신자가 처음 수행한 투척을 후속 투척으로 정의하였다. 133회 중 132회에서 후속 투척이 관찰되었으며, 수신부터 투척까지 걸린 시간의 중앙값은 1.27초였다. 전체 성공 패스 중 53회(39.8%)는 1초 이내 투척으로 이어졌고, 27회(20.3%)는 후속 투척이 상대 적중으로 연결되었다. [그림 5](b)는 후속 투척이 관찰된 132회의 누적분포로, 가로축의 시간 이내에 후속 투척한 비율을

세로축에 나타낸다. [그림 5](c)의 세 비율은 모두 성공 패스 133회를 분모로 한다. 후속 투척에는 재패스도 포함된다.

전방 패스의 높은 비중과 수신 이후의 투척은 후방에서 전방 팀원으로 공을 전달하고 행동을 이어 가는 패턴을 보여준다. 본 연구에서는 이러한 공 전달과 후속 행동의 연계를 협력적 상호작용으로 해석한다. 패스 자체에는 직접 보상이 없지만 팀 보상은 존재하므로, 팀 목표를 공유하는 학습 조건에서 나타난 연계 행동이다.

5.4 상대 무적 상태에 따른 투척 빈도

생존한 상대 중 무적 에이전트가 없는 경우와 한 명 이상인 경우(비무적 상대가 함께 있는 경우 포함)를 비교하였다. 각 조건에서 100경기의 총 투척 수를 총 공 보유 시간(초)으로 나누고 60을 곱해 공 보유 1분당 투척 횟수를 산출하였다.



[그림 6] 상대 무적 상태별 공 보유 시간당 투척 빈도

[Fig. 6] Throw frequency per minute of ball possession by opponent invincibility

[표 6] 상대 무적 상태별 공 보유 시간과 전체 투척

[Table 6] Ball-possession duration and all throws by opponent invincibility

상대 상태	공 보유 시간	투척 수	1분당 투척
무적 상대 없음	725.05초	698회	57.76회
한 명 이상 무적	927.19초	189회	12.23회

[그림 6]과 [표 6]에서 상대 중 한 명 이상이 무적일 때는 무적 상대가 없을 때보다 공 보유 시간이 길었지만, 공 보유 1분당 전체 투척 횟수(패스 포함)는 적었다. [그림 6]의 오차 막대는 경기를 단위로 10,000회 재표집해 산출한 백분위 부트스트랩 95% 신뢰구간이다.

6. 결론 및 향후 연구

본 연구는 Unity 기반 2대2 피구에서 모방학습 없이 MA-POCA와 자기대전으로 약 1억 스텝을 학습하고, 정책을 고정해 100경기를 평가하였다. 공 보유 팀에 따라 전·후방 공간 활용이, 상대의 무적 상태에 따라 투척 빈도가 달라졌다(RQ1). 성공 패스 133회 중 100회가 전방 패스였으며, 패스에 직접 보상을 주지 않아도 공 전달과 후속 투척이 관찰되었다(RQ2). 이는 소규모 팀 게임에서 시연 없이도 상황에 반응하고 팀원과 행동을 연계하는 AI를 학습시킬 수 있음을 보여준다.

자기대전은 플레이 시연을 수집하거나 별도의 상대 AI를 구현하지 않고도 학습에 필요한 경쟁 경험을 제공한다. 개발자는 관찰·행동·보상을 설계하고 구체적인 행동 조합은 학습에 맡길 수 있기에, 플레이 데이터가 부족한 소규모 게임에서 상황별 행동 규칙 작성의 부담을 줄일 수 있다(RQ3).

한계는 행동 규칙 작성의 부담이 학습 문제의 설계와 검증 부담으로 옮겨 간다는 데 있다. 개발자는 보상과 관찰이 게임의 목표 및 상황 판단을 지원하는지, 보상이 의도하지 않은 행동을 유도하는지 점검해야 한다. 본 연구에서는 경기장 5개를 병렬로 실행했으며, 앞서 명시한 장비에서 약 1억 스텝 학습에 대략 80시간이 소요되었다. 이는 구현뿐 아니라 반복 학습과 평가에 필요한 시간도 개발 일정에 고려해야 함을 보여준다.

이번 실험은 하나의 학습 정책에서 나타난 행동을 분석한 사례다. 규칙 기반 AI나 다른 학습 방식과의 직접 비교는 후속 과제로 남아 있다. 에이전트 수, 경기장 크기, 아이템 규칙 등이 바뀌는 경우에는 관찰 구조와 정책의 적응성도 확인해야 한다.

후속 연구에서는 초기·중기·최종 체크포인트를 동일한 상대 조건에서 비교해 행동과 팀원 간 협력의 변화를 분석하고, 실제 플레이와 설문을 통해 도전성·자연스러움·협력성·재미를 평가하고자 한다(RQ4). 또한 보상 설계, 커리큘럼 전환 조건, 모방학습 병행 여부의 영향과 경기 조건 변화에 대한 정책의 적응성을 검토할 수 있다.

Ng 등이 제시한 잠재함수 기반 보상 변환의 최적 정책 보존 조건 [15]을 참고해 보조 보상이 게임 목표에 부합하는지 검토할 수 있다. 또한 보상 항목을 하나씩 제거해 행동과 경기 성과의 변화를 비교할 수 있다. 시연 확보가 가능하다면 GAIL [11] 등 모방학습과 비교해 시연 수집 비용, 학습 효율, 행동 특성을 평가할 수 있다.

소규모 게임에서의 활용 전망은 경기 기록을 통한 행동 분석을 개발자의 설계·평가 과정에 연결하는 데 있다. 학습 정책의 경기 성과와 플레이어 경험을 함께 검토하고, 확인된 문제를 관찰·보상·게임 규칙의 개선으로 연결한다면 소규모 팀 게임에 적합한 AI를 개발하는 데 활용할 수 있다.

References

[1] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems*, vol. 30, Long Beach, CA, USA, Dec. 4-9, 2017, pp. 6380-6391.

[2] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, "A survey and critique of multiagent deep reinforcement learning," *Autonomous Agents and Multi-Agent Systems*, vol. 33, no. 6, pp. 750-797, Nov. 2019, doi: 10.1007/s10458-019-09421-1.

[3] B. Baker et al., "Emergent tool use from multi-agent autocurricula," in International Conference on Learning Representations, Virtual Conference, Apr. 26-May 1, 2020.

[4] T. Rashid, M. Samvelyan, C. Schroeder de Witt, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *35th International Conference on Machine Learning*, vol. 80, Stockholm, Sweden, Jul. 10-15, 2018, pp. 4295-4304.

[5] C. Yu et al., "The surprising effectiveness of PPO in cooperative multi-agent games," in *Advances in Neural Information Processing Systems*, vol. 35, New Orleans, LA, USA, and Virtual Conference, Nov. 28-Dec. 9, 2022, pp. 24611-24624, doi: 10.52202/068431-1787.

[6] A. Cohen et al., "On the use and misuse of absorbing states in multi-agent reinforcement learning," Jun. 2022, arXiv:2111.05992, doi: 10.48550/arXiv.2111.05992.

[7] M. Jaderberg et al., "Human-level performance in 3D multiplayer games with population-based reinforcement learning," *Science*, vol. 364, no. 6443, pp. 859-865, May 2019, doi: 10.1126/science.aau6249.

[8] A. Juliani et al., "Unity: A general platform for intelligent agents," May 2020, arXiv:1809.02627, doi: 10.48550/arXiv.1809.02627.

[9] V.-P. Berges, "ML-Agents plays DodgeBall," *unity.com*, https://unity.com/blog/engine-platform/ml-agents-plays-dodgeball (accessed Oct. 1, 2026).

[10] F. Lin, S. Huang, T. Pearce, W. Chen, and W.-W. Tu, "TiZero: Mastering multi-agent football with curriculum learning and self-play," in *22nd International Conference on Autonomous Agents and Multiagent Systems*, London, United Kingdom, May 29-Jun. 2, 2023, pp. 67-76.

[11] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems*, vol. 29, Barcelona, Spain, Dec. 5-10, 2016, pp. 4572-4580.

[12] L. V. R. Cavadas, S. Melo, T. C. Kohwalter, and E. Clua, "Using provenance data and imitation learning to train human-like bots," *Entertainment Computing*, vol. 48, Art. no. 100603, Jan. 2024, doi: 10.1016/j.entcom.2023.100603.

[13] Unity Technologies, "Training configuration file," *github.com*, https://github.com/Unity-Technologies/ml-agents/blob/release_22_docs/docs/Training-Configuration-File.md (accessed Oct. 1, 2026).

[14] Unity Technologies, "Training ML-Agents," *github.com*, https://github.com/Unity-Technologies/ml-agents/blob/release_22_docs/docs/Training-ML-Agents.md (accessed Oct. 1, 2026).

[15] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Sixteenth International Conference on Machine Learning*, Bled, Slovenia, Jun. 27-30, 1999, pp. 278-287.