“Top Gun: Maverick - Dog Fight”

(공중전 에이전트 성능 최적화를 위한 강화학습 설계 요소 비교 분석)

2025.12.07

탑건: 매버릭

서강대학교 20211214 한민성

E-Mail: alwayzhan@gmail.com

목차

1. 프로젝트 주제 및 목표

프로젝트 주제

강화학습을 활용한 자율 공중전(Dogfight) 에이전트 프로젝트이다. 2D 공중전 시뮬레이션 환경에서, 강화학습 에이전트의 **학습 효율을 극대화하는 설계 요소(환경·보상·알고리즘·모델·학습전략)** 를 비교 분석하는 프로젝트이다.

프로젝트 목표

1. 강화학습이 학습 가능한 최적 환경(난이도·보상·규칙) 설계 기준 규명한다.
2. PPO vs SAC, 네트워크 유닛 수, 학습 전략(HARD, Curriculum 등)의 효율성 정량 비교한다.
3. 객관적이고 공정한 평가 체계(Win Rate 기반)를 구축하여 최적 조합 도출한다.

Dogfight: 두 전투기가 서로를 충분히 볼 수 있는 거리에서 벌이는 근접 공중전

관련 영상 1: https://www.youtube.com/watch?v=huFZYu_YK6Q, 관련 영상 2: https://www.youtube.com/watch?v=NgjQkFIZaco

1. 프로젝트 주제 및 목표| 핵심 방향

“승리“보다 “최적화”에 집중

- 단순히 적을 이기는 모델을 만드는 것은 쉽다. (최종본의 세팅 값에서 난이도를 낮추면 되고, step 수를 늘리면 됩니다.)
- 어떤 환경 및 학습 전략 하에서 에이전트가 가장 효율적으로 학습하는가를 분석하는 데 초점을 맞췄다.

비교 분석

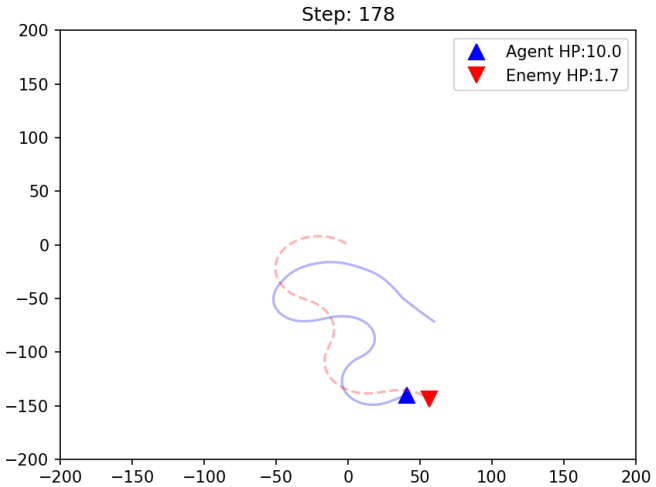
- **알고리즘**: PPO vs SAC 알고리즘 비교
- **모델 구조**: 신경망 유닛 수(64 vs 128 vs 256)에 따른 수렴 속도와 효율성 비교
- **학습 전략**: 여러 가지 고정 환경 학습과 커리큘럼 학습의 최종 수렴 안정성 비교

독자적인 시뮬레이션 환경 구축

1) State, Action, Reward 설계 2) Agent 및 Enemy 설정 설계 3) 실험 환경 설계

2. 시뮬레이션 환경 및 규칙 | 시뮬레이션 개요

- **환경**: 2 Gymnasium 기반의 2D 공중전 시뮬레이터 구축
- **물리 모델**: 고정익기(Fixed-wing) 역학을 단순화하여, 일정 속력으로 방향을 전환함
- **구성**: 항공 역학(속도, 선회율, 등), 적기 전술 로직(Rule-based AI), 보상 함수 등 실험에 필요한 모든 변수를 직접 하드 코딩하며 설계하여 실험 변수를 완벽히 통제함



선회율: 전투기가 방향을 얼마나 빠르게 바꾸는가

Lock-on: 위 게임에서는 Lock-on 조건이 되면 상대의 체력이 매 step마다 깎이게 된다. (매 스텝마다 기관총을 고쏜)

2. 시뮬레이션 환경 및 규칙

전투 및 승리 규칙

- **승리 조건**: 적기를 향해 조준원(Lock-on) 안에서 일정 시간 동안 유지하여 상대의 체력을 0으로 만들면 승리
- **공격 매커니즘**
  - 거리: 사거리(LOCK_DIST) 이내 진입
  - 각도: 적기가 내 기수 전방 특정(LOCK_ANGLE) 이내에 위치
  - 판정: 위 두 조건을 만족하면 'Lock-on' 상태가 되며 적에게 지속적인 데미지를 줌

시뮬레이션 세부 규칙

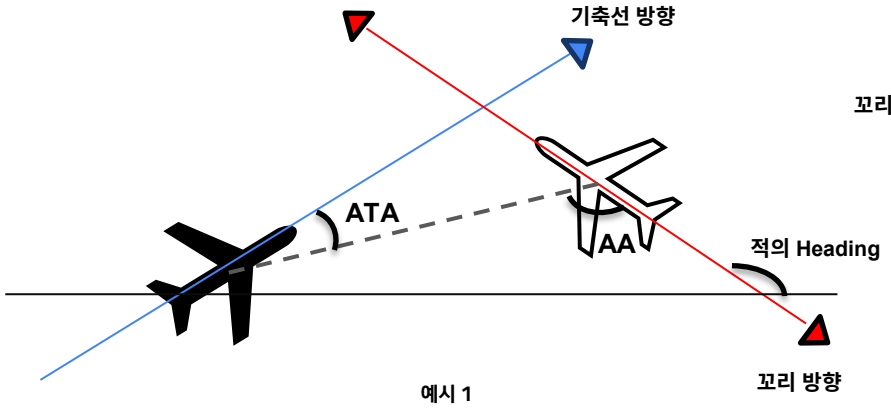
- **시작점**: 확률에 따라 (1) 'Agent가 유리한 위치(적의 뒤에서 시작)', (2) 'Agent와 적 모두 Random 위치' 에서 시작
- **적의 전략**: 확률에 따라 (1) 'PURSUIT(추격 알고리즘)', 2) 'HYBRID(회피기동+추격 알고리즘)', (3) 'LEFT/RIGHT(노이즈를 포함한 각도로 좌/우 방향으로 원을 그리는 알고리즘)' 중에 하나의 전략을 선택하여 진행
- **종료 조건**: (1) '체력이 0이 되어 격추', (2) '충돌', (3) '맵 밖으로 이탈', (4) '일정 step 도달'

회피 기동 알고리즘: 적에게 공격당할 수 있는 특정 조건(ex. Lock-on)이 되었을 때 급격하게 방향을 틀어 적의 공격 범위에서 벗어나는 알고리즘

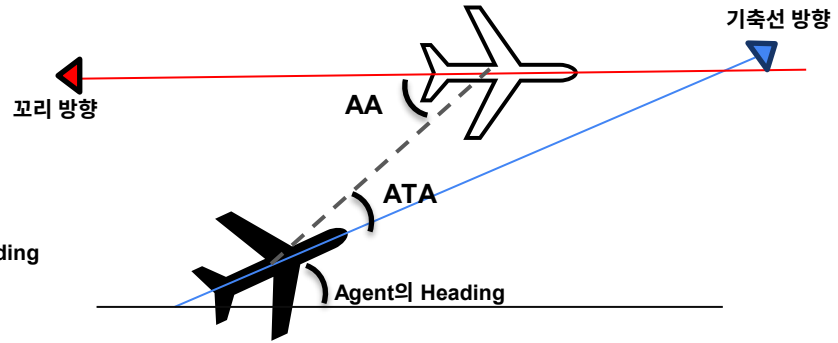
2. 시뮬레이션 환경 및 규칙

기하학적 관계

- **Heading:** 기수의 각도 (향하고 있는 방향의 절대적 각도)
- **ATA(Antenna Train Angle)**: 내 기축선과 적기 사이의 각도 (Lock-on 조준 여부 판단)
- **AA(Aspect Angle)**: 적기의 꼬리 방향과 나 사이의 각도 (내가 적의 뒤에 있는지 판단)
- **핵심 목표**: ATA를 줄여 조준(공격) 가능하도록 하고, AA를 줄여 꼬리를 잡은 상태(공격에 유리하며 적에게 공격을 당하지 않는 위치)를 유지하는 것



예시 1



예시 2

선회율: 전투기가 방향을 얼마나 빠르게 바꾸는가

3. 환경 설계 | 수많은 시행착오를 통한 '학습 가능성 ' 의 발견

초기 가설 및 접근

- 이상적인 **ATA, AA 각도**와 적기에 **추격(Pursuit) 로직**을 적용하면, Agent가 생존을 위해 수많은 시행착오(Step)를 겪으며 스스로 이상적인 추격 및 회피기동 로직을 찾아낼 것이라 기대함
- 이상적인 설정 값과 " 강한 적을 상대로 학습해야 더 강한 에이전트가 나온다 " 는 전제로 접근

실패 사례 분석

- **현상:** Agent가 유의미한 행동(회피/반격)을 시도해 보기도 전에 적기에게 즉시 격추당함
- **결과:** 보상을 전혀 획득하지 못하는 **학습 불능** 상태가 지속됨
- **추가 시도:** Agent의 ATA, AA, 체력 등 다른 설정 값을 유리하게 조정해 보았으나, 적기의 완벽한 추격 로직 앞에서는 무용지물이었음

핵심 인사이트

- **"어려운 환경이 능사가 아니다."**
  - 물리적으로 극복 불가능한 상황(Physical Impossibility)에서는 학습이 일어나지 않음
- **환경 설계의 중요성:**
  - Agent가 행동과 결과 사이의 **인과관계**를 파악할 수 있도록 '**최소한의 생존 및 반격 기회**'를 보장하는 환경 설계가 선행되어야 함을 규명함

3. 환경 설계 | 다차원 변수 최적화를 통한 Baseline 학습 환경 구축

1. 학습 성패를 결정짓는 3대 요소와 복합적인 상호작용

- **복합적 인과관계**
  - **'게임 규칙', '난이도', '보상 체계'**는 독립적이지 않고 유기적으로 연결되어 성능을 결정함
  - **예시**: 보상이 완벽해도 적의 난이도가 높으면 학습이 불가능, 난이도가 적절해도 보상 설계가 잘못되면 이상 행동 발생

2. 환경 최적화를 위한 3대 고려 사항

지수적으로 많은 변수 조합 속에서 최적의 설정을 찾기 위해 다음 3가지 기준을 적용

- **학습 가능성 (Learnability)**: 에이전트가 포기하지 않고 인과관계를 학습할 수 있는가?
- **실험 가능성 (Feasibility)**: 제한된 자원(Time, Step) 내에서 수렴 가능한가? (CPU 사용)
- **변별력 (Discriminative Power)**: 알고리즘 및 주요 변수 변화에 따른 성능 차이를 명확히 구분할 수 있는가?

3. 학습 가능 영역 탐색 및 Baseline 확립

- **반복적인 실험**
  - 위 3가지 기준을 충족하는 변수 조합을 찾기 위해 파라미터 튜닝을 반복하여 에이전트가 유의미한 성장을 보이는 지점 탐색
- **표준화**
  - 단순한 추측이 아닌 데이터 기반 검증을 통해, 표준 시뮬레이션 규칙 및 Baseline 환경을 최종 확립

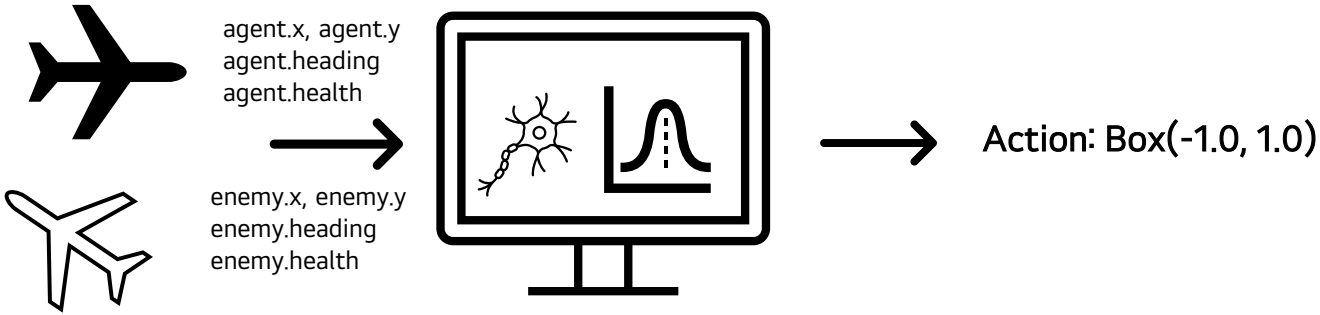
4. 강화학습 설계 | State, Action

1. State (8-dim Observation)

- 에이전트가 상황을 판단하기 위해 필요한 최소한의 필수 정보만을 선별하여 8차원 벡터로 구성함
  - Agnet의 정보: x좌표, y좌표, Heading(기수 방향 각도), HP(체력)
  - 적의 정보: x좌표, y좌표, Heading(기수 방향 각도), HP(체력)

2. Action Space (Continous Control)

- **Type**: Box(-1.0, 1.0) (연속적인 실수값 제어)
- **Action**: **선회율(Trun Rate)**을 제어하여 비행 궤적을 결정
  - **-1.0**: 최대 좌선회
  - **+1.0**: 최대 우선회
  - **0.0**: 직진



4. 강화학습 설계 | Reward Shaping

1. 보상 함수 구성

에이전트가 전술적 우외 점유 → 공격 → 격추의 과정을 학습할 수 있도록 4가지 카테고리로 보상을 세분화

- **전술 기동 유도**
  - **R_position**: 적의 꼬리를 점유하고, 기수를 적에게 향할 때 보상
  - **R_closure**: 유효 사거리(30~100) 내로 접근 시 보상, 충돌 위험 거리(<30) 진입 시 패널티
- **핵심 행동**
  - **R_gunsnap**: 조준선(Lock-on) 안에 적을 포착하고 유지하는 행동에 보상, 반대로 당할 때는 패널티
- **최종 목표**
  - **R_win/R_lose**: 적기 격추 시 큰 보상, Agent 격추 당할 시 패널티
- **안전 장치**
  - **R_crush/R_boundary**: 적기 충돌 또는 맵 이탈 시 강력한 패널티

$$R_{total} = (R_{pos} + R_{close} + R_{gun} + R_{outcome} + R_{safety}) \times 0.1$$

4. 강화학습 설계 | Reward Shaping

2. 시행착오 및 정교한 계산 기반 최적화

초기 설계에서 발생한 이상 행동을 교정하기 위해 수많은 시뮬레이션을 거쳐 보상 밸런스를 최적화함

- **자살 에이전트 문제 해결**
  - **문제**: 초기에는 이탈 패널티가 피격 누적 고통과 비슷하여 에이전트가 고통을 피하기 위해 시뮬레이션이 시작되거나 피격이 시작될 때 바로 맵을 이탈하는 최적해를 찾음
  - **해결**: 이탈(-110) < 피격 누적(-50) + 격추(-50) 부등식이 성립하도록 패널티를 재설계하여, 아무리 불리해도 끝까지 생존하여 싸우는 것이 수학적으로 이득임을 학습시킴

- **행동 유도 강화**
  - **문제**: 초기에는 R_position 가중치가 낮아, 에이전트가 '적의 뒤를 잡는 것 ' 의 전략적 중요성을 인지하지 못하고 공격 기회를 만들지 못함
  - **해결**: 위치 점유 보상의 가중치를 대폭 상향하여, Positioning을 우선적으로 학습하도록 유도

- **학습 안정성 확보**
  - **해결**: 보상 값의 절댓값이 너무 커서 발생하는 신경망 발산 문제를 방지하기 위해, 최종 리워드에 0.1의 스케일링을 적용하여 학습 수렴 속도를 개선

5. 강화학습 알고리즘 선정 및 Hyperparameter 설정

1. 최종 알고리즘 선정: SAC (Soft Actor-Critic)

연속 제어 환경인 공중전 시뮬레이션에 적합한 알고리즘을 선정하기 위해, On-policy 알고리즘인 PPO와 Off-policy 알고리즘인 SAC를 비교 분석

선정 근거 (실험 결과 참고)

- **Sample Efficiency (샘플 효율성)**
  - **PPO**: 데이터를 일회성으로 소비하고 폐기하므로, Sparse Reward 환경에서 학습 속도가 느림
  - **SAC**: Replay Buffer를 통해 과거의 성공적이 격추 경험을 반복 학습하므로, 데이터가 귀한 공중전 환경에서 매우 유리

- **Exploration (탐험)**
  - **SAC**: **최대 엔트로피**를 통해 에이전트가 고정된 패턴에 갇히지 않고 다양한 회피 및 공격 기동을 시도하도록 유도하여 로컬 최적해 탈출에 유리함

5. 강화학습 알고리즘 선정 및 Hyperparameter 설정

2. Hyperparameter 설정

Parameter	Value	비교 실험
Algorithm	PPO / SAC	O
Network Arch	[64, 64], [128, 128], [256, 256]	O
Seed (Training)	1, 2, 3	O
Step	2000000	고정
Learning Rate	3e-4	고정
Batch Size	256	고정
Buffer Size	1000000	고정
Gamma	0.99	고정
Tau	0.005	고정

6. 실험 셋업 | 학습 환경

1. Multi-Processing

- SubprocVecEnv를 활용하여 10개의 환경을 병렬 구동, 데이터 수집을 10배 향상시켜 실험 효율성을 극대화함
- 한정된 자원 CPU로 최소한의 시간으로 최대한 많은 실험을 하기 위한 세팅

2. 난이도 정의 (자세한 수치는 다음 페이지 표 참고)

- EASY / HARD 의미: **절대적인 기준이 아니라**, 수많은 시뮬레이션을 통해 찾아낸 '**학습 가능한 범위**' **내에서의 상대적인 수준**을 의미함
    - **EASY 특징**: 기초 기동 학습을 위한 설계로 적보다 Lock-On 각도가 크며 90%의 확률로 적의 뒤에서 에피소드를 시작한다.
    - **HARD 특징**: 적과 Lock-On 각도가 같으며, EASY보다는 Agent가 더 RANDOM한 위치에서 시작한다. 또한 적은 EASY보다 높은 확률로 고급 기동 전략을 선보인다.
- **RANDOM 특징**: 100%의 확률로 Random 위치에서 에피소드를 시작하며, Agent와 같은 선회율을 갖고 고급 기동 전략(추격+회피)을 선보인다.
- Curriculum_V1 / Curriculum_V2의 의미: 적의 난이도가 선형보간 방식으로 점차 올라가서 HARD 수준까지 올라감
    - **Curriculum_V1 특징**: 200만 step 기준으로, 0~100만 step까지 선형보간 방식으로 적의 난이도가 EASY에서 HARD로 올라간다.
    - **Curriculum_V2 특징**: V1 실험 이후에 난이도 변화에 따른 Agent의 혼란을 막고자 EASY보다 조금 더 어려운 난이도(앞으로 NORMAL로 칭함)에서 HARD에 도달하게끔 하였다.

6. 실험 셋업 | 학습 환경

3. 적(Enemy) 난이도 설정

- 난이도 비교:
  - EASY < Curriculum_V1 < Curriculum_V2 < HARD <<<<<<< RANDOM

Parameter \ 난이도	EASY	HARD	Curriculum_V1	Curriculum_V2	RANDOM
Agent / Enemy의 Speed	2.0 / 1.5	2.0 / 1.5	2.0 / 1.5	2.0 / 1.5	2.0 / 1.5
Agent Lock-on 각도	35	30	35 → 30	30	30
Enemy Lock-on 각도	30	30	30	30	30
선회율(t_rate)	0.05	0.05	0.05	0.05	0.1
시작 위치(랜덤 / 적 뒤)	0.9 / 0.1	0.7 / 0.3	0.9 / 0.1 → 0.7 / 0.3	0.8 / 0.2 → 0.7 / 0.3	1 / 0
적의 움직임 (Hybrid / Pursuit / L+R)	0.2 / 0.3 / 0.5	0.4 / 0.3 / 0.3	0.2 / 0.3 / 0.5 → 0.4 / 0.3 / 0.3	0.3 / 0.3 / 0.4 → 0.4 / 0.3 / 0.3	1 / 0 / 0

6. 실험 셋업 | 학습 환경

4. 학습 재현성 및 통계적 검증

- 독립 시행

- 단일 학습 결과의 우연성을 배제하기 위해 모든 실험군(알고리즘, 신경망 유닛 수, 난이도)에 대해 3개의 고정된 Seed(1, 2, 3) 를 사용하여 독립적인 학습을 3회 반복함 (2000000 step 고정)

- 신뢰도 확보

- 최종 결과는 3회 실험의 평균과 표준오차로 산출되어 성능 안정성을 검증 (이후, 실험 결과 파트에서 확인 가능)

6. 실험 셋업 | 평가 지표

1. 평가 철학

- **객관성 확보**: 학습 중 얻는 보상(Reward)뿐만 아니라 추가적으로 객관적인 지표가 추가로 필요하다.
- **전략**: 학습 진행도와 무관하게, 고정된 기준(Standardized Benchmark)에서 에이전트의 실질적인 전투력을 검증해야 한다.

2. 핵심 지표

- Reward
- Win Rate
  - **평가 주기 (Checkpoints):** 학습의 세밀한 변화 추이를 추적하기 위해 매 20000 step마다 모델 Checkpoint를 저장하여 평가 시 해당 시점의 승률 측정
  - **엄격한 승리 기준**: Max Step 초과나 충돌 등 단순 생존은 배제하고, 오직 적을 격추했을 때만 승리로 인정
  - **통계적 유의성**: 50회의 반복 교전을 통해 우연성을 배제하고 승률의 신뢰도 확보

3. 재현성 및 공정성 – Win Rate

- Episode-Seed Mapping
  - 평가 시 각 에피소드 번호에 해당하는 **Random Seed를 고정 (예시: Ep 1 → Seed 1, EP 2 → Seed 2, ...)**
  - **효과**: PPO, SAC, EASY 등 모든 모델이 똑같은 50개의 시나리오를 수행하게 하여 공정한 비교가 보장

6. 실험 셋업 | 평가 지표

4. 평가 난이도 최적화

- 난이도 보정

- **문제**: 평가 난이도가 너무 낮으면 승률이 100% 수렴, 난이도가 너무 높으면 0%에 수렴하여 성능 증가율을 파악할 수 없음
- **해결:** 수많은 사전 실험을 통해, 모델의 성장 추이를 민감하게 보여줄 수 있는 EVAL_LV.1를 도출하여 Baseline으로 설정

- 계층적 평가

- **목적**: 상위권 모델 간의 미세한 성능 차이를 검증하면서도 실험의 효율성을 확보하기 위함
- **Step 1**: 모든 모델을 **EVAL_LV.1**에서 평가하여 기본적인 성능 수렴 여부를 확인
- **Step 2**: **EVAL_LV.1**에서 성능이 유사하거나 포화 상태에 도달한 최상위 모델들에 한해, 난이도를 상향한 **EVAL_LV.2** 환경에서 추가 평가를 수행하여 변별력을 확보함

Parameter \ 난이도	EVAL_LV.1	EVAL_LV.2
Agent / Enemy의 Speed	2.0 / 1.5	2.0 / 1.5
Agent Lock-on 각도	35	30
Enemy Lock-on 각도	25	25
선회율(t_rate)	0.04	0.05
시작 위치(랜덤 / 적 뒤)	0.3 / 0.7	0.5 / 0.5
적의 움직임 (Hybrid / Pursuit / L+R)	0.3 / 0.3 / 0.4	0.4 / 0.2 / 0.4

6. 실험 셋업 | 비교 실험 설계

동일한 평가 환경 하에서 단일 변수/난이도 만을 변경하며 성능 차이를 분석

- **알고리즘**: PPO vs SAC 알고리즘 비교
  - 희소 보상 및 연속 제어 환경에서의 샘플 효율성과 최종 수렴 성능 차이 확인
- **모델 구조**: 신경망 유닛 수(64 vs 128 vs 256)에 따른 수렴 속도와 효율성 비교
  - 모델 복잡도에 따른 학습 속도와 실시간 추론 효율성 간의 트레이드오프 분석
- **학습 전략**: 여러 가지 고정 환경 학습과 커리큘럼 학습 각각의 최종 수렴 안정성 비교
  - 고정 환경(Fixed): 목표 환경 직접 노출을 통한 학습 효율 검증
  - 커리큘럼 환경(Curriculum): 난이도 전이 과정에서의 적응 비용(cost)와 초기 학습 안정성 간의 상관관계 분석

실험 \ 비교군	비교군 1	비교군 2	비교군 3	비교군 4	비교군 5
Algorithm (알고리즘)	PPO	**SAC**			
Neural Network (네트워크)	[64, 64]	**[128, 128]**	[256, 256]		
Difficulty (난이도)	EASY	**HARD**	Curriculum_V1	Curriculum_V2	RANDOM

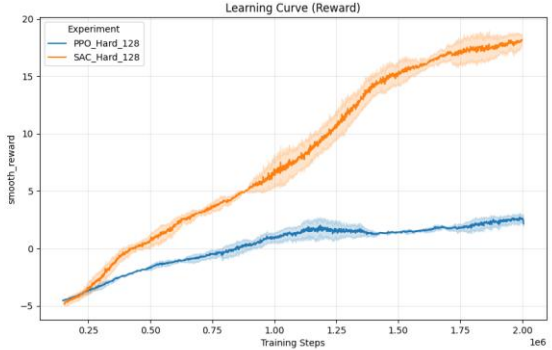
표에서 음영처리 된 것은 Baseline
→ 알고리즘을 평가할 때는 [128, 128], HARD로 통일
→ 난이도를 평가할 때는 SAC, [128, 128]로 통일

모델 Checkpoint가 필요하시거나 추가 문의 사항 있으시면 메일로 연락주시면 감사드리겠습니다.
E-Mail: alwayzhan@gmail.com

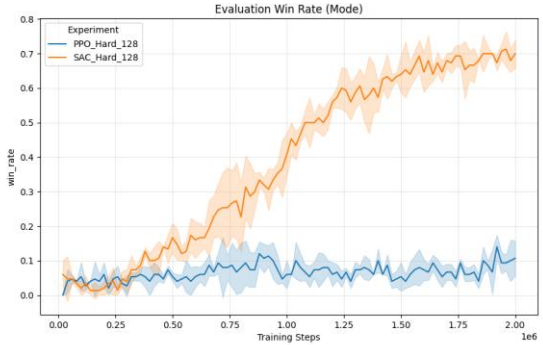
7. 실험 결과 1 | 알고리즘 성능 비교

그래프의 실선: 각 Seed(1, 2, 3)으로 학습된 모델들의 평균
그래프의 음영: 각 Seed(1, 2, 3)으로 학습된 모델들의 신뢰구간

1. PPO vs SAC 학습 곡선 분석



Reward Graph



Win Rate Graph – (평가: EVAL_LV.1)

- 성능 격차 확인
  - **SAC**: 동일한 200만 Step 학습 시, SAC는 평균 보상이 10점 후반대, 승률 70%에 수렴함
  - **PPO**: 반면 PPO는 평균 보상 5점 미만, 승률 10% 대에 머무르며, 유의미한 정책을 학습하지 못함
- 원인 분석
  - **PPO** (On-Policy): 데이터를 일회성으로 소비하여, 희소한 '격추 성공 ' 경험을 충분히 학습하지 못함
  - **SAC** (Off-Policy): Replay Buffer를 통해 과거의 성공 데이터를 지속적으로 재학습하여, 복잡한 전술 기동을 효과적으로 습득

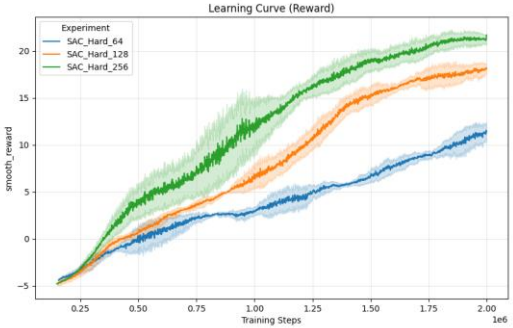
2. 결론: SAC 채택

- 데이터의 효율성이 성능 결정의 핵심 요인임을 확인하고 SAC를 Baseline으로 결정

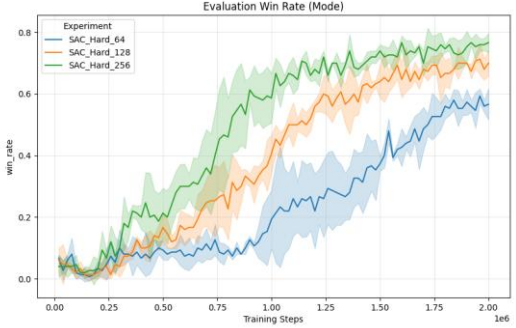
7. 실험 결과 2 | 네트워크 구조 최적화

그래프의 실선: 각 Seed(1, 2, 3)으로 학습된 모델들의 평균
그래프의 음영: 각 Seed(1, 2, 3)으로 학습된 모델들의 신뢰구간

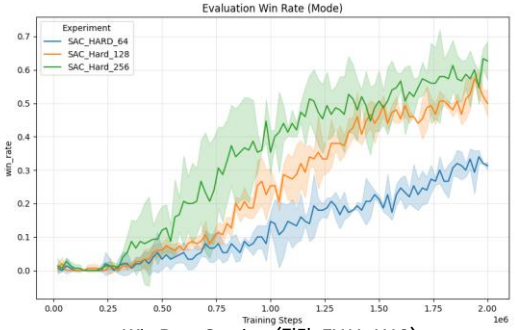
1. Neuron 수(64 vs 128 vs 256)에 따른 성능 비교



Reward Graph



Win Rate Graph – (평가: EVAL_LV.1)



Win Rate Graph – (평가: EVAL_LV.2)

- **수렴 속도 vs 최종 성능**
  - **256 Unit**: 128 Unit 보다 빠르게 수렴하고 성능이 가장 좋음
  - **128 Unit**: 256 Unit에 비해 느리게 수렴하나 최종 도달한 성능은 256 Unit과 큰 차이가 없음
  - **64 Unit**: 256 Unit, 128 Unit에 비해 정교한 움직임을 보여주지 못하며, Reward 및 Win Rate의 차이가 눈에 띄게 존재함
- **Trade-off 분석**
  - 모델의 파라미터 수가 늘어날 수록 연산 비용이 증가함 (64Unit: 약 50분, 128 Unit: 약 100분, 256 Unit: 약 180분)
  - 최종 성능이 비슷하다면, 동일 시간 안에 많은 시뮬레이션, 많은 실험을 할 수 있는 가벼운 모델을 선택

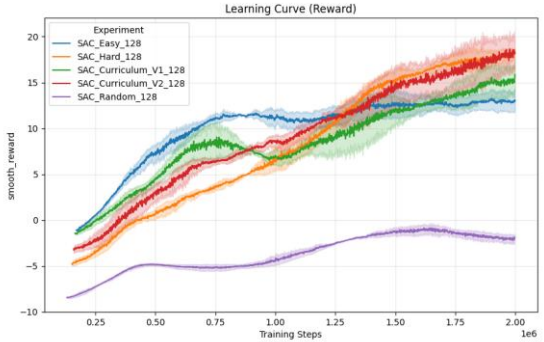
2. 결론: 128 Unit 채택

- 불필요한 연산 자원 낭비를 막고 실험 효율성을 높이기 위해 128Unit 구조를 Baseline으로 결정

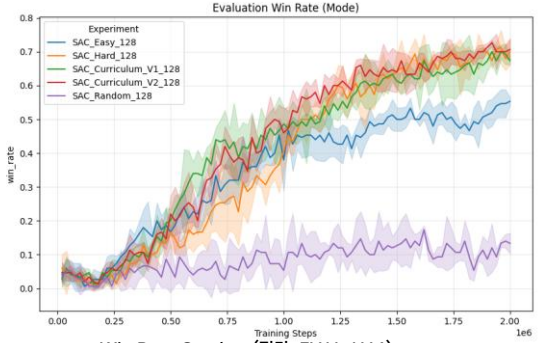
7. 실험 결과 3 | 학습 전략 분석

그래프의 실선: 각 Seed(1, 2, 3)으로 학습된 모델들의 평균
그래프의 음영: 각 Seed(1, 2, 3)으로 학습된 모델들의 신뢰구간

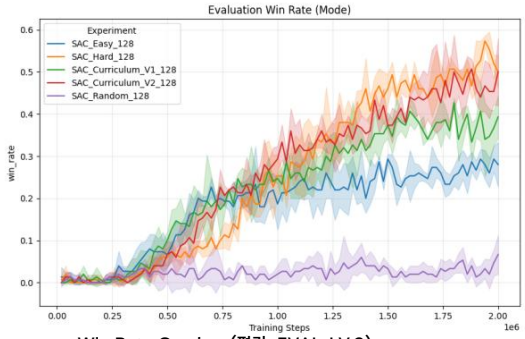
고정 환경과 점진적 커리큘럼 학습 비교 (EASY – HARD – RANDOM - Curriculum_V1 - Curriculum_V2)



Reward Graph



Win Rate Graph – (평가: EVAL_LV.1)



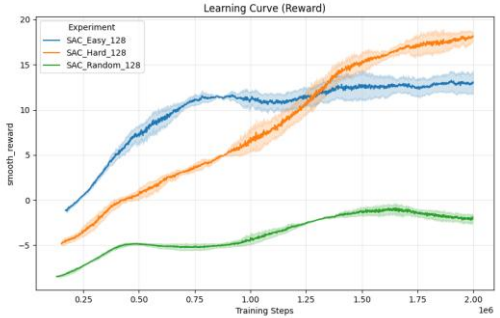
Win Rate Graph – (평가: EVAL_LV.2)

- 난이도 및 주요 설정 값 (15, 16 slides 참고)
  - EASY < Curriculum_V1 < Curriculum_V2 < HARD <<<<<<< RANDOM

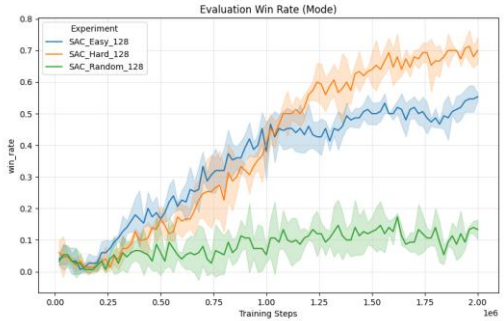
- **학습 전략 비교**
  - (1) EASY / HARD vs RANDOM
  - (2) Curriculum_V1의 한계
  - (3) Curriculum_V2 최적화 (HARD와의 비교)

7. 실험 결과 3 | 학습 전략 비교 (1): EASY/HARD vs RANDOM

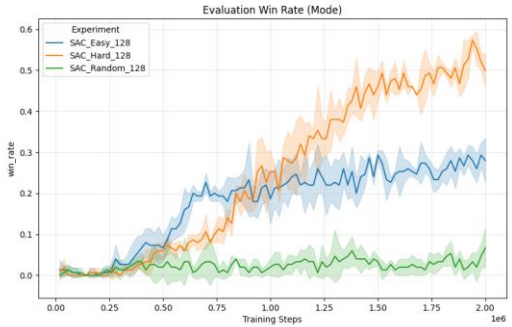
그래프의 실선: 각 Seed(1, 2, 3)으로 학습된 모델들의 평균
그래프의 음영: 각 Seed(1, 2, 3)으로 학습된 모델들의 신뢰구간



Reward Graph



Win Rate Graph – (평가: EVAL_LV.1)



Win Rate Graph – (평가: EVAL_LV.2)

RANDOM 모드의 학습 실패

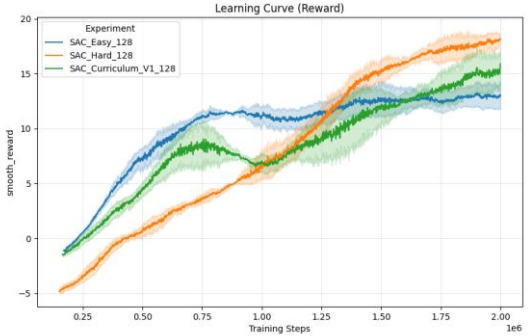
- **결과**: 보상 점수와 승률이 0에 수렴하며 학습이 전혀 이루어지지 않음
- **원인 분석**
  - 에이전트가 아무런 가이드 없이 회피 기동을 하는 적을 상대로 유효 타격을 입히는 것은 확률적으로 불가능에 가까움
  - **인사이트**: 학습 초기에는 인과관계를 파악할 수 있는 **'최소한의 성공 경험** ' 을 제공하는 것이 필수적임을 증명함

최소한의 성공 경험 제공과 적절한 난이도 제공의 사이

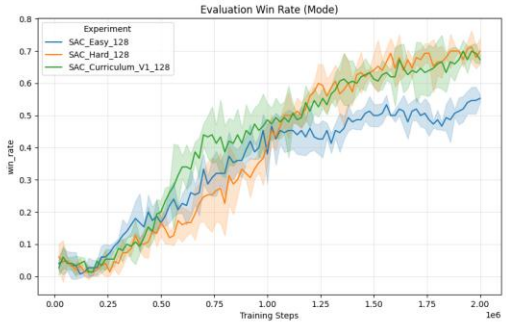
- RANDOM 모드처럼 최소한의 성공 경험을 줄 수 없는 난이도를 주는 것은 학습 불능 상태로 만든다. 반대로 큰 어려움 없이 성공 경험을 할 수 있도록 하는 것은 어느 순간 학습 진행을 멈추게 한다. 이 사이에서 적절한 조절이 필요하다.

7. 실험 결과 3 | 학습 전략 비교 (2): Curriculum_V1의 한계

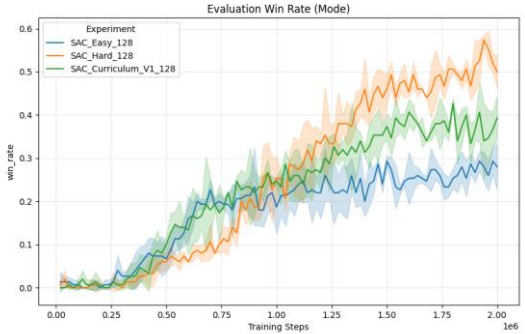
그래프의 실선: 각 Seed(1, 2, 3)으로 학습된 모델들의 평균
그래프의 음영: 각 Seed(1, 2, 3)으로 학습된 모델들의 신뢰구간



Reward Graph



Win Rate Graph – (평가: EVAL_LV.1)



Win Rate Graph – (평가: EVAL_LV.2)

1. Curriculum_V1 설계

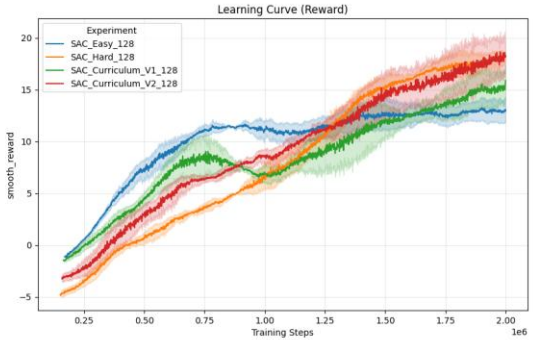
- **방식**: 0 Step(EASY) → 100만 Step(HARD)까지 난이도를 선형 보간 방식으로 증가시킴.
- **기대 효과:** 학습 초기에는 EASY 장점을 가져가면서 후반에는 HARD의 수렴성을 기대함

2. 성능 하락 현상 분석 (그래프: V1의 급격한 V자 곡선)

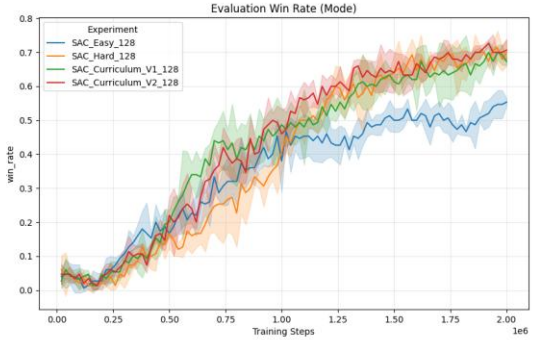
- **현상:** 초반에는 높은 승률을 보였으나, 100만 Step(HARD 도달 시점) 부근에서 성능이 급격히 붕괴됨
- **원인:**
  - 초반 EASY 환경에서 습득한 단순 추적 습관이 고난도 환경에서는 통하지 않음
  - 기존 정책을 폐기하고 재학습하는 과정에서 큰 Cost가 발생하여, 최종적으로 HARD Baseline의 성능을 넘어서지 못함

7. 실험 결과 3 | 학습 전략 비교 (3): Curriculum_V2 최적화

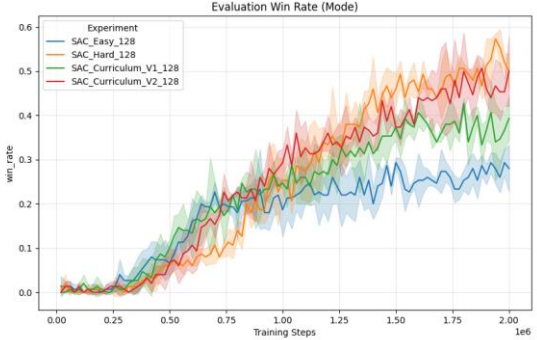
그래프의 실선: 각 Seed(1, 2, 3)으로 학습된 모델들의 평균
그래프의 음영: 각 Seed(1, 2, 3)으로 학습된 모델들의 신뢰구간



Reward Graph



Win Rate Graph – (평가: EVAL_LV.1)



Win Rate Graph – (평가: EVAL_LV.2)

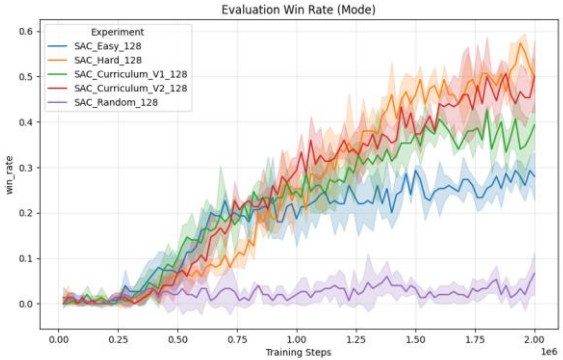
Curriculum_V2 개선 전략

- **Start Difficulty 상향**
  - 시작 난이도를 EASY보다 높게 설정(NORMAL)하여, 초기 학습과 최종 목표(HARD) 간의 격차를 축소
- **Slower Ramp**
  - 난이도 상승 기울기를 완만하게 조절하여 에이전트가 환경 변화에 부드럽게 적응하도록 유도
  - **가설**: 이를 통해 V1의 성능 하락을 제거하고, 초기부터 후반까지 안정적인 우상향 학습 곡선을 그릴 것이다.

7. 실험 결과 3 | 학습 전략 비교 (3): Curriculum_V2 최적화

그래프의 실선: 각 Seed(1, 2, 3)으로 학습된 모델들의 평균
그래프의 음영: 각 Seed(1, 2, 3)으로 학습된 모델들의 신뢰구간

[그래프: EVAL_LV.2에서의 V2 vs HARD 승률 비교]

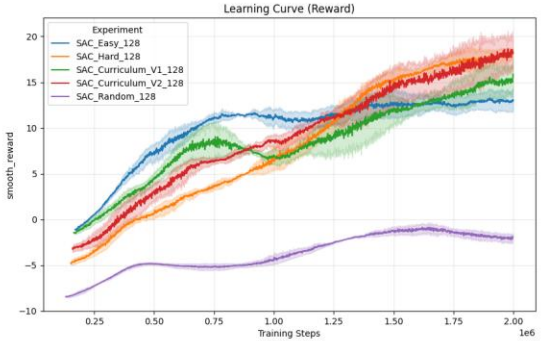


Win Rate Graph – (평가: EVAL_LV.2)

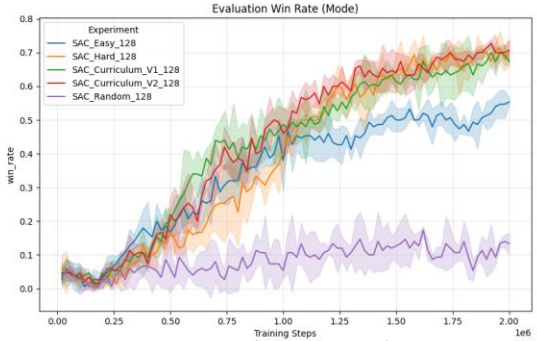
- 1. 결과 관찰
  - **후반 역전 및 수렴:** 학습 후반부에서 HARD 모델이 V2보다 소폭 높은 승률을 기록하다가 최종적으로 비슷해짐
- 2. 심층 분석: "경험의 총량 차이 "
  - 고난도 경험의 차이
    - **HARD**: V2에 비해 고난도 적(Hybrid)과 어려운 위치(랜덤한 위치)에 비교적 더 많이 노출됨. 어려운 상황을 해결하는 경험의 총량이 V2보다 많음
    - **Curriculum_V2**: 초기에는 쉬운 적 위주로 학습하다가 점진적으로 어려운 적의 비중을 늘렸기 때문에, 고난도 패턴을 학습할 절대적인 시간이 HARD 모델보다 부족
  - 결론
    - Curriculum_V2는 안정적인 성장을 보장하지만, 최고 난이도 환경에 대한 숙련도를 극한으로 끌어올리는 데는 고정 환경 학습(HARD)보다 시간이 더 걸릴 수 있다.

7. 실험 결과 3 | 학습 전략 비교 (3): Curriculum_V2 최적화

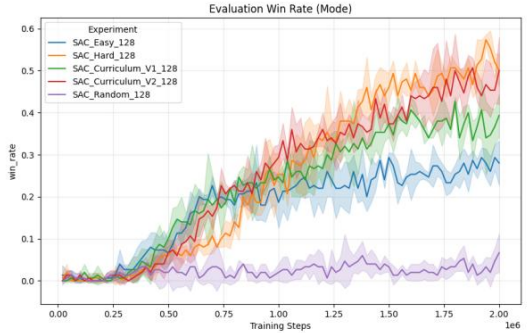
그래프의 실선: 각 Seed(1, 2, 3)으로 학습된 모델들의 평균
그래프의 음영: 각 Seed(1, 2, 3)으로 학습된 모델들의 신뢰구간



Reward Graph



Win Rate Graph – (평가: EVAL_LV.1)



Win Rate Graph – (평가: EVAL_LV.2)

구분	HARD	Curriculum_V2
강점	빠른 목표 환경 적응 속도	초기 학습 안정성
약점	초기 학습 불안정성	고난도 도달 시간 필요
결론	극한의 특정 테스크 수행에 적합	범용적인 테스크 수행에 적합

모델 Checkpoint가 필요하시거나 추가 문의 사항 있으시면 메일로 연락주시면 감사드리겠습니다.
E-Mail: alwayzhan@gmail.com

8. 토의 및 결론 | 실험 결과에 대한 생각

0. 알고리즘 특성 및 모델 용량의 영향

본 실험에서 PPO와 SAC를 비교했을 때, SAC가 더 우수한 성능을 보였다. 또한 신경망의 은닉층 유닛수를 확장했을 때 유의미한 성능 향상이 관찰되었다. 이는 2D 기동과 적기 추적이라는 복잡한 State를 해석하기 위해서는 에이전트가 환경 정보를 충분히 추상화 할 수 있는 표현력이 필수적임을 시사한다. 유닛 수가 부족할 경우에는 복잡한 비행 역학을 온전히 모델링하지 못하는 과소적합 현상이 발생하여 학습 성능이 특정 구간에 정체되는 한계를 보였다.

1. 학습 안정성과 특화 성능의 Trade-Off

Curriculum_V2는 초기 학습 실패 확률을 최소화하며 안정적인 우상향 그래프를 그렸다. 이는 에이전트가 쉬운 과제에서 얻은 '기초 지식 ' 을 바탕으로 어려운 과제로 전이하는 과정이 유효했음을 뜻한다. 반면 HARD는 초반 0% 성공률이라는 것을 극복해야 했다. 이를 돌파한 후에는 목표 환경에 최적화된 정책을 빠르게 수렴시켰다. 이는 모든 상황을 습득하기보다 필요한 상황에만 집중했기 때문으로 분석된다.

2. 범용성의 대가

Curriculum_V2가 HARD보다 최종 수렴 도달 시간이 늦거나 특정 구간 성능이 낮은 이유는, 에이전트의 신경망이 NORMAL과 HARD의 특성을 모두 수용할 수 있는 파라미터를 찾아야 했기 때문인 것으로 해석된다. 반대로 HARD는 Curriculum_V2에 비해 LV.2 평가 기준(고난도)에 더욱 Fit하게 학습하였기 때문에 해당 특정 환경에서는 더 높은 점수를 기록할 수 있었다.

3. "시작부터 어렵게" 전략의 유효성

일반적인 강화학습 이론에서는 Sparse Reward 문제로 인해 Curriculum Learning을 권장한다. 그러나 위 실험의 HARD 결과는 환경의 복잡도가 에이전트의 탐색 능력 범위를 벗어나지 않는다면, 처음부터 목표 난이도에 노출시키는 것이 학습 효율 면에서 더 유리할 수 있음을 보여주었다. 이는 HARD의 환경을 설계할 때부터 에이전트에게 최소한의 성공 경험을 제공하도록 구성하였기 때문에 이같은 결론이 도출한 것으로 보인다.

8. 토의 및 결론 | 향후 연구 과제 및 개선 사항

위 프로젝트의 한계를 극복하고 에이전트의 성능과 범용성을 극대화하기 위해, 컴퓨팅 자원이 충분히 확보된다는 가정 하에 다음과 같은 심화 연구를 제안한다.

1. 학습 효율성 극대호를 위한 지식 증류 기법 도입 (Teacher-Student)

Curriculum_V2의 안정성과 HARD의 성능이라는 두 가지 장점을 모두 취하기 위해 'Teacher-Student' 학습 구조를 도입한다.
(Curriculum_V2와 HARD로 예시를 들었지만, 이것이 아니더라도 비교적 안정적인 모델과 우수한 성능을 가진 모델로 시도할 만한 가치가 있을 것으로 보인다)

- **배경**: 범용적인 Curriculum_V2는 수렴하는 데 비교적 시간이 걸리며, HARD 모델은 초기 학습의 유연성이 떨어진다는 단점이 있다.
- **방법**: 먼저 Curriculum_V2로 안정적으로 학습된 모델을 Teacher로 둔다. 이후 HARD 환경에 배치된 초기 Student 모델이 Teacher 모델의 행동 패턴을 모방하도록 손실 함수를 설계하여 학습시킨다.
- **기대 효과:** Student 모델은 Teacher 모델의 가이드를 통해 초반 탐색 비용을 줄일 수 있다. 결과적으로 고난도 환경에 빠르게 적응하면서도, 최종적으로는 Teacher보다 뛰어난 고성능 에이전트를 확보할 수 있을 것이다.

2. 자기 복제를 통한 난이도 자동 향상

현재의 한계가 정해져 있는 적을 넘어, 일정 수준 이상 학습된 에이전트의 정책을 적 기체에 복사하여 경쟁시키는 전략을 도입한다.

- 이는 알파고와 같이 에이전트가 자신을 이기기 위해 끊임없이 더 강력한 전략을 개발하는 효과를 유도한다.
- 이를 통해 사전에 정의된 난이도를 넘어, 예측 불가능한 전술을 구사하는 적을 상대로도 승리할 수 있는 초고성능 에이전트로의 진화를 기대할 수 있다.

8. 토의 및 결론 | 향후 연구 과제 및 개선 사항

3. 시뮬레이션 고도화 및 확장

충분한 학습 Step을 확보하여, 단순한 특정 난이도 정복이 아닌 무작위 환경에서도 완벽한 대응이 가능한 수준까지 일반화 성능을 끌어올리는 것을 목표로 한다. 이를 위해 시뮬레이션 환경을 다음과 같이 다각도로 확장하고자 한다.

- **State Dimension 확장** : 현재의 위치, 속도 정보 외에 연료량, 총알 수, 등 현실적인 제약 조건을 State에 추가하여 시뮬레이션의 정밀도를 높인다. 이를 통해 에이전트가 단순 기동 뿐만 아니라 자원 관리 능력까지 학습하도록 유도한다.
- **Multi-Agent 전투 환경 구축**: 현재의 1:1 공중전 시나리오를 넘어, 다수의 아군과 적군이 협력 및 경쟁하는 환경으로 확장한다. 이를 통해 에이전트는 개별 전투 기술 뿐만 아니라, 편대 비행 및 전술적 협력 능력을 가질 수 있다.
- **Unity 기반의 시각화**: Unity 엔진을 연동하여 에이전트의 비행 궤적과 교전 상황을 고품질 그래픽으로 실시간 구현한다. 이는 에이전트의 기동을 직관적으로 분석할 수 있게 한다.

모델 Checkpoint가 필요하시거나 추가 문의 사항 있으시면 메일로 연락주시면 감사드리겠습니다.
E-Mail: alwayzhan@gmail.com

Thank you

긴 보고서 내용 읽어 주셔서 감사드립니다.

모델 Checkpoint가 필요하시거나 추가 문의 사항 있으시면 메일로 연락주시면 감사드리겠습니다.
E-Mail: alwayzhan@gmail.com