Simulating Point-and-Click Behavior

in

Implicit Adversarial Environment

2015143535 박진형

2013145069 심규철

2016147512 이현우

지도교수 : 이병주 교수님

목차

1. 연구 주제

본 연구에서는 적대적인 agent가 존재하는 상황에서의 인간의 Point-and-Click Behavior를 강화학습으로 모델링하는 연구를 진행했다. 또한, 적대적인 agent가 존재하는 환경에서 학습시킨 agent를 적대적인 agent가 존재하지 않는 환경에서 학습시킨 기존 모델과 정성적, 정량적으로 비교하였고, 더 나아가 agent의 Point-and-Click Task 수행 과정에 Decision Making Skill 관련 Human Factor들과 Motor Execution 관련 Human Factor들이 어떤 영향을 미치는 지에 대한 연구를 ablation study로 진행하였다.

2. 배경

HCI (Human Computer Interaction) 분야에서 많은 연구들의 목표는 User Performance를 수학적으로 Modeling 하는 것이다. User Performance란 인간과 컴퓨터의 상호작용이 얼마나 효과적으로 이루어지고 있는지를 나타내는 말인데, User Performance를 수학적인 모델로 모델링할 수 있다면 크게 2가지 효과를 기대할 수 있다. 먼저, 인간과 컴퓨터를 연결하는 Interface가 바뀌었을 때 User의 Performance가 어떻게 변화할지 예측할 수 있게 되고, 반대로 User의 성격이 바뀌었을 때 동일한 Interface에서 User Performance가 어떻게 달라질지 또한 예측할 수 있게 된다. 따라서 이를 확장시킨다면 User Performance를 극대화하는 최적화된 Interface Design을 찾아내거나 User Performance를 극대화하는 방향으로 가이드라인을 제시할 수 있게 된다.

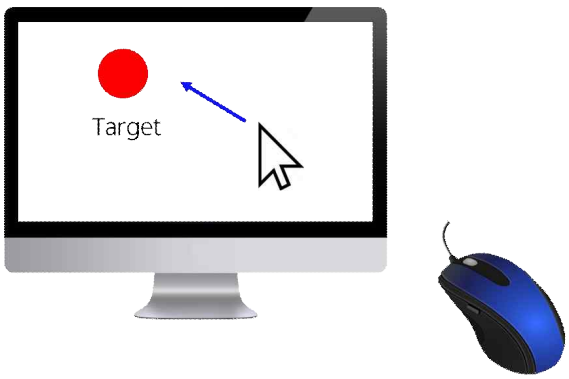


Fig 1. Point-and-Click Task

인간과 컴퓨터의 상호작용 중에서 현재 가장 많이 수행되는 기본적이면서도, 가장 중요한 Task는 “Point-and-Click” Task 인데, Point-and-Click Task는 어떤 인테페이스에서 사용자가 클릭하거나 취득하고자 하는 target이 있을 때 커서(cursor)를 target으로 움직이는 tracking process와 그 target을 클릭하는 clicking process로 나눌 수 있다. 따라서 HCI의 많은 연구들에서는 Point and Click Task를 수행하는 과정을 모델링하고 있으며, 이를 Point-and-Click Behavior Modeling 이라고 한다. 이러한 Point-and-Click Behavior Modeling을 통해 Point-and-Click Task를 수행하는 과정을 최적화시킬 수 있고 최적화된 인터페이스를 설계할 수 있게 된다. 또한, Point and Click Task를 수행하는 Agent를 학습시킬 수 있다면, 인터페이스 최적화 외에도 스스로 point-and-click task를 수행하는 agent를 통해 특정 작업을 자동화시킬 수 있을 것이고, 인간과 경쟁하는 Agent를 개발하는 등 다양한 분야에서 활용할 수 있을 것이다.

3. 기존 연구들과의 차별점

본 연구는 Point and Click Behavior를 Modeling 하고자 한 기존 연구들과 크게 3가지 측면에서 차별점을 지닌다.

(1) 통합적인 모델 제안

Point-and-Click Task는 Tracking Process와 Clicking Process로 나뉘는데 거의 모든 선행연구들이 개별적인 Process만을 모델링하고 있다. 하지만 Tracking Process와 Clicking Process는 주어진 환경 내에서 상호연관되어있기 때문에 이들을 각각 모델링한 모델들을 단순히 합칠 경우, 각 process 들이 서로 미치는 영향이나 그 과정에서 새로 생기는 변수들을 고려할 수가 없다. 즉, Point-and-Click Behavior를 모델링하기 위해서는 개별적인 Process들을 모델링할 것이 아니라 이러한 사항들을 종합적으로 고려한 통합모델이 필요하다.

(2) 인간의 특성을 직접적으로 고려한 모델링

Point-and-Click Behavior를 모델링할 때는 인간의 신체적인 특성이나 해당 task를 수행할 때 인간의 인지사고과정 등을 고려해서 모델링 해야하는 반면, 기존의 선행연구들에서는 이러한 부분들을 직접적으로 고려하고 있지 않다. Point-and-Click task를 수행할 때 영향을 주는 요소들은 Target의 위치/속력과 현재 Cursor의 위치 외에도 cursor를 움직이는 사람의 손목, 인터페이스를 인식하는 사람의 시각기관, 눈동자 직임, 그리고 이를 인식해서 의사결정을 내리는 인지과정 등 수많은 인간적인 요소들이 개입되어있다. 본 연구에서는 이러한 요소들을 유일하게 고려했던 선행 연구 [1]에서 사용한 human factor들을 활용하여 각 human factor가 인간의 Point-and-Click 수행과정에 어떤 영향을 미치는지 검증하였다.

(3) Multiple Agent가 적대적으로 경쟁하는 실제 세계의 특징 반영

Fig 2. 동일한 Target을 두고 여러 agent가 Point-and-Click Task를 경쟁하는 경우

앞서 설명한 두 가지 한계점을 보완한 Point-and-Click Simulation Model [1] 이 최근에 제안되었지만 이 역시 한계점이 존재한다. 해당 연구 [1] 에서는 Target을 클릭하고자 하는 Agent가 1명일 때를 기준으로 모델링 했는데, 실제로 Point-and-Click를 수행하는 많은 환경에서는 동일한 target을 취득하고

자 하는 agent가 여러 명인 경우들이 많다. 게임이나 선착순 클릭과 같이 고도의 정확성과 클릭 시점을 요구하는 환경의 경우, 동일한 target을 두고 여러 agent들이 경쟁하게 될 때 (비록 상대방의 움직임을 직접적으로는 보지 못하더라도) 인간은 상대방의 존재를 의식하여 잠재적으로 본인의 의사결정 및 Point-and-Click Task 수행과정에 영향을 미치게 된다. 이러한 적대적인 agent들이 존재할 때에는 Point-and-Click Behavior가 single agent일 때의 behavior와 달라질 것이고, 실제 세계에서의 User Performance를 모델링하기 위해서는 이러한 Adversarial Agent가 존재하는 Environment에서의 behavior를 모델링해야 한다.

4. 연구 방법

(1) 연구 문제 정의

본 연구에서는 2개의 연구 소주제를 정의했다. 첫 번째 연구 주제는 Human Factor 값이 서로 동등한 2개의 agent를 적대적으로 강화학습시키는 연구, 두 번째로는 Human Factor 값이 서로 다른 2개의 agent를 적대적으로 강화학습시키는 연구를 진행했다.

Table 1: This table shows the free parameters that describe the simulated user's cognitive and behavioural characteristics. The parameter values were set by referring to previous studies assuming an average user.

Variable	Description	Value	Ref	Module
T_p	Planning time interval	0.1 s	[11]	Motor control
n_v	Motor noise constant (parallel)	0.2	[44]	Upper limb
n_p	Motor noise constant (perpendicular)	0.02	[44]	Upper limb
l_{se}	Shoulder-to-elbow length	25.7 cm	[40]	Upper limb
l_{ew}	Elbow-to-wrist length	25.7 cm	[54]	Upper limb
l_{wh}	Wrist-to-hand length	6.43 cm	[40]	Upper limb
σ_v	Width of likelihood of visual speed perception	0.15	[60]	Visual perception
$f_{gain}()$	Mouse acceleration function	OS X 10.12	[14]	Mouse
c_σ	Precision of internal clock	0.09015	[41, 53]	Click action
c_μ	Implicit aim point	0.185	[41, 53]	Click action
ν	Drift rate	19.931	[41, 53]	Click action
δ	Visual encoding precision limit	0.399	[41, 53]	Click action

Fig 3. 본 연구에서 사용한 Human Factor 값 (기존 연구 [1]에서 사용한 값 차용)

Agent 1

(Improved **Decision-Making Skill** Agent)

변수		Value	Module
c_μ	Precision of internal clock	0.185 → 0.3	Click action
c_σ	Implicit aim point	0.09015 → 0.06	Click action
ν	Drift rate	19.931 → 40	Click action
δ	Visual encoding precision limit	0.399 → 0.25	Click action

Agent 2

(Improved **Motor Execution** Agent)

변수		Value	Module
n_v	Motor noise constant (parallel)	0.2 → 0.24	Upper limb
n_p	Motor noise constant (perpendicular)	0.02 → 0.024	Upper limb
σ_v	Width of likelihood of visual speed perception	0.15 → 0.18	Visual Perception

Fig 4. 2번째 연구 문제에서 각 agent의 human factor를 조정한 방법

먼저 Human Factor란, agent가 선택한 action을 수행하여 다음 state로 업데이트되는 과정에 인간적인 요소를 반영하기 위해 사용하는 상수값이다. (해당 상수 값들이 구체적으로 언제 어떻게 반영되는지는 4-(2) 적대적 강화학습 설계 참고) Human Factor는 총 5개의 영역 – Visual Perception Module, Motor Control Module, Upper Limb Module, Click Action Module, Mouse Module – 으로 구성되어 있으며 총 11개의 Factor 들이 존재한다 (Fig. 3). 각 Human Factor의 구체적인 값들은 평균적인 유저를 모델링했던 기존 연구들의 결과에 따라 그 값이 결정되었다.

첫 번째 연구 문제의 경우, 2개의 agent들의 Human Factor 값을 기존 연구 [1] 와 모두 동일한 값을 사용(통제변인)해서 적대적으로 강화학습시켰다. 그 후, 적대적으로 강화학습 시킨 두 agent에는 어떤 차이점이 있는지 검증하고, 두 agent 중 하나의 agent를 적대적으로 강화학습시키지 않았던 기존 모델(agent) [1]와 비교함으로써 과연 적대적으로 강화학습시키는 것이 optimal policy (종속변인)에 어떤 변화를 주는지 분석하는 연구를 진행했다

두 번째 연구 문제의 경우, 각 agent의 Human Factor 값을 서로 다르게 조정시킨 후 각 Human Factor가 Point-and-Click Task 수행 과정 및 optimal policy에 어떤 영향을 미치는지 분석하는 실험을 ablation study로 진행했다. 첫 번째 agent의 경우 Decision-Making Skill이 향상되도록 이와 관련된 human factor 값들을 조정했고, 두 번째 agent의 경우 Motor Execution 능력이 향상되도록 관련 human factor 값들을 조정했다 (Fig 4). Decision-Making Skill이란 클릭 의사결정을 정확하고 정밀하게 내리는 능력을 말하는데, 이 능력을 강화하기 위해 관련된 Human Factor들을 조정할 경우 Point-and-Click Task 중 Clicking Process의 수행 능력을 높이는 효과를 기대할 수 있다. 반대로, Motor Execution Skill이란 주어진 타겟의 속도를 정밀하게 지각한 후 계획한 움직임을 강건하게 실행하는 능력을 말하기 때문에 이와 관련된 Human Factor 값들을 조정하는 것은 Tracking Process의 수행능력을 조정하는 것과 동일한 효과를 기대할 수 있다. 이렇게 서로 다른 능력을 강화시킨 두 agent를 적대적으로 강화학습시킴으로써 두 Human Factor 영역 중 어떤 영역이 Point-and-Click Task에 더 큰 영향을 미치는지 분석하는 실험을 진행했다.

(2) 적대적 강화학습 환경 설계

본 연구에서는 앞서 기술한 연구 문제들을 해결하기 위해 적대적 강화학습 환경을 Fig 5와 같이 설계하였다. 먼저, Fig 5.에 기술된 강화학습 환경을 single agent를 기준으로 (왼쪽 절반에 대해서만) 설명하면 다음과 같다.

- **State 및 Environment**

먼저, 본 연구에서 agent란 Point-and-Click Task를 수행하는 마우스 커서(cursor)를 의미하는데, agent가 속한 환경(environment)은 agent가 클릭을 목표로 하는 target들을 생성하는 장소가 된다 (Fig 6). 따라서 agent의 현재 상태(state)는 총 11개의 변수 – 커서의 위치($p_{cursor-x}$, $p_{cursor-y}$)와 속력

($v_{cursor-x}$, $v_{cursor-y}$), 타겟의 위치($p_{target-x}$, $p_{target-y}$)와 속력($v_{target-x}$, $v_{target-y}$), 타겟의 반지름(r_{target}), 그리고 커서를 마우스로 움직이는 사람 손목의 위치(p_{hand-x}, p_{hand-y}) - 로 정의된다. 이 때, 직관적으로

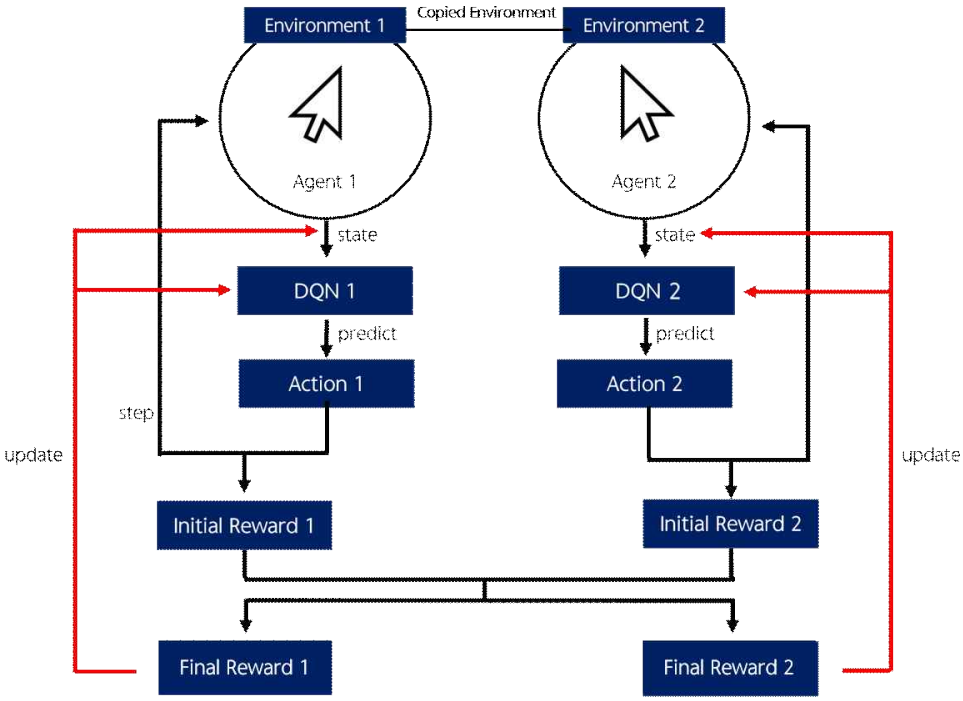


Fig 5. 본 연구에서 설계한 적대적 강화학습 환경

cursor의 현재 상태(state)에 상대방의 cursor에 대한 정보(위치, 속력)가 추가되어야 한다고 생각될 수 있다. 하지만 실제 세계에서 대부분의 경우, 한 유저가 상대 유저의 커서의 위치나 속력을 직접적으로 알 수 있는 경우는 없고, 상대방의 존재를 의식하는 유일한 경로는 결과를 통해서만 (예를 들면 유저가 특정 타겟을 클릭하기 이전에 상대방이 타겟을 클릭해서 해당 타겟이 유저의 시야에서 먼저 사라지는 것과 같은 현상을 통해서만) 상대방을 간접적으로 의식할 수 있다. 따라서 이러한 사실에 입각하여 본 연구에서는 현재 상태를 정의하는 변수를 위와 같이 11개로 설정하였다.

- **Deep-Q-Network 및 Action**

이러한 state 정보는 Deep-Q-Network (이하 DQN)의 입력값으로 들어오고, DQN은 해당 state에서 취할 수 있는 가장 좋은 action을 선택하게 된다. 즉, 이러한 Point-and-Click Task는 한 시점의 state에 의해서만 다음 시점의 state가 결정되는 Markov Decision Process로 규명할 수 있다. DQN이 선택하는 action space는 총 50가지의 action들로 구성된 이산적인 action space이며 각 action은 클릭 시도 여부를 의미하는 Click Decision(K)와 Prediction Horizon(T_h)의 순서쌍 (K, T_h)으로 정의된다.

먼저, 해당 시점에서 클릭을 시도할지($K=1$) 혹은 시도하지 않을지($K=0$)에 따라 50개의 action 들은 각각 25개씩으로 나뉘게 된다. 각 25개의 action들은 agent가 cursor를 움직일 때 고려하는 Prediction Horizon(T_h)의 값에 따라 다시 나뉘는데, 여기서 Prediction Horizon 이란 agent가 cursor를

움직일 때 얼마나 먼 미래까지를 내다보아서 의사결정을 내릴지에 대한 시간적인 시야 범위를 말한다. 이 Prediction Horizon은 0초에서 2.4초까지 0.1초 간격으로 총 25개의 이산적인 값을 가지는데 $\{T_h \mid 0 \leq T_h \leq 2.4\}$, $T_h = 2.4$일 경우, 현재보다 2.4초 이후의 미래까지 타겟이 어떻게 움직일지를 예측하여 커서를 어느 방향으로 얼마만큼 움직일지 결정하겠다는 것을 의미하고 $T_h = 0$일 경우 미래를 내다보지 않고 현재 target의 위치만을 고려하여 커서를 움직이겠다는 것을 의미한다.

하지만 이렇게 의사결정을 내리는 것과 실제로 이를 행하는 것 사이에는 인간적인 요소가 개입되어야 한다. 따라서, 이렇게 선택된 action 값 (K, T_h)은 앞서 기술한 11개의 Human Factor 값들과 함께 고려되어서 최종적으로 클릭 시도 여부와 cursor의 위치와 속력 값 등이 업데이트된다.

- **Reward 및 Policy Update**

DQN을 통해 선택된 action을 기반으로 state가 업데이트되면, 그에 따라 reward가 부여된다. Reward는 Click의 성공 여부에 따라 부여되는 Click Reward와 해당 target을 클릭하는데까지 유저가 손목에 들인 노력을 정량화하는 Motor Effort로 구성되는데, 구체적인 reward의 산정방식은 Fig 7과 Fig 8에 기술되어있다. 이렇게 산정된 Reward에 따라서 DQN의 가중치 값들은 업데이트되고 결국 이 과정을 수백만개의 학습 episode 동안 반복하면서 agent가 Point-and-Click Task를 수행하는 최적의 정책(optimal policy)를 찾아나가도록 강화학습시켰다.

이렇게 optimal policy를 찾아나가는 각 agent 2개를 적대적으로 동시에 학습시키기 위해 본 연구에서는 다음과 같이 environment, reward, episode termination 기준을 정의했다.

① Environment

먼저 본 연구에서 state에는 상대방 커서의 위치와 속력 정보가 포함되지 않는다고 기술했다. 따라서 본 연구에서는 서로 같은 target을 생성하는 2개의 environment를 생성했다. 각 agent는 서로 다른 환경에서 학습을 하지만, 2개의 환경은 매 학습 episode마다 Fig 6과 같이 서로 동일한 target을 생성하기 때문에 두 agent는 마치 동일한 환경에서 동일한 target을 두고 경쟁하는 것처럼 학습시킬 수 있다. 이렇게 학습 환경을 서로 분리할 수 있는 이유는 상대 cursor의 위치와 속력이 서로의 의사결정에 미치지 않기 때문 (state에 포함되지 않기 때문) 이며, 각 환경에서 agent들이 target을 클릭한 결과들만 함께 종합하여 reward를 최종적으로 차등 부여하면 되기 때문이다. 따라서, 매 학습 episode 마다 각 agent가 target을 클릭 시도하는데 걸린 시간과 클릭의 성공 여부를 각각 독립적인 환경에서 먼저 1차적으로 측정했다.

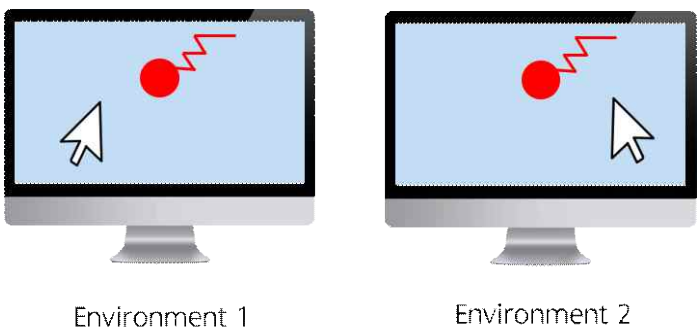


Fig 6. 적대적 강화학습을 수행한 환경

② Reward

그 후에는 앞서 측정한 결과를 기준으로 reward를 산정하는데 reward는 Fig 7의 예시에서 기술한 방식처럼 산정하였다. 먼저 Fig 7의 경우 각 agent가 만약 혼자 있었다면 (상대 agent가 서로 존재하지 않는 상황이었다면) 두 agent는 모두 클릭에 성공했으므로 두 agent에게는 각각 긍정적인 reward를 부여해야 한다. 하지만, 두 Agent 2가 Agent 1보다 먼저 클릭에 성공했기 때문에 사실 3초 이후에는 target이 사라진 것이나 다름없으므로 Agent 1은 사실상 클릭에 실패한 것과 같다. (모델 학습 시 별도의 2개의 환경을 구성한 이유는 개별 agent의 클릭 성공 여부와 클릭하는데 걸린 시간을 1차적으로 쉽게 측정하기 위한 것이었다.) 따라서 이 경우, Agent 1에게는 클릭을 성공했을 때 받는 기존의 긍정적인 reward(R++)에서 일부 reward를 차감시켜서 reward를 차등 부여(R+)하는 방식으로 최종적으로 부여할 reward를 결정했다.

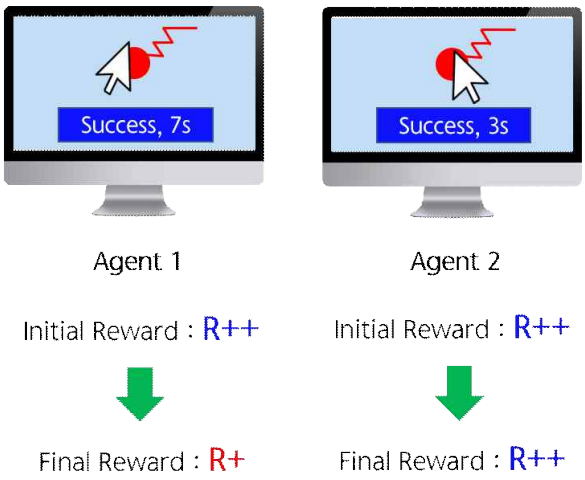


Fig 7. 적대적 강화학습 시 Reward를 차등 부여한 방식

Reward를 산정하는 구체적인 식은 Fig 8에 기술되어있다. Reward는 앞서 기술했듯이 클릭의 성공 여부에 따라 부여되는 click reward와 target을 선택하기 위해 유저가 손목에 들인 노력의 정도를 정량화하는 motor effort로 구성되는데, motor effort가 크면 클수록 많은 노력을 들여서 target을 취득한 것이므로 motor effort는 음수값이 되도록 했다. 그 후, Reward = Click Reward + Effort Reward로 설정했다.

Case	Click Reward	Motor Effort	Example
Case 1 (Click 시도 이전)	0	$-\sum_{t=t_0+T_p}^{t_0+T_p+T_h} \left\|\dot{\hat{v}}_h[t]\right\|$	-
Case 2 (빠르게 Click 성공)	14	$-\sum_{t=t_0+T_p}^{t_0+T_p+T_h} \left\|\dot{\hat{v}}_h[t]\right\|$	상대보다 먼저 클릭해서 성공한 경우, 상대가 먼저 실패한 후에 내가 성공한 경우
Case 3 (늦게 Click 성공)	9	$-\sum_{t=t_0+T_p}^{t_0+T_p+T_h} \left\|\dot{\hat{v}}_h[t]\right\|$	상대가 먼저 성공한 이후에 내가 성공한 경우
Case 4 (Click Fail)	-1	$-\sum_{t=t_0+T_p}^{t_0+T_p+T_h} \left\|\dot{\hat{v}}_h[t]\right\|$	시점에 관계없이 클릭에 실패한 경우

Fig 8. Reward 산정식 (Reward = Click Reward – Motor Effort)

Click Reward는 크게 4가지 case로 나뉘는데, 클릭을 시도하기 전에는 0을, 클릭을 실패했을 경우에는 -1이라는 Click Reward를 부여했다. 두 agent가 모두 클릭에 성공한 경우, 먼저 클릭에 성공한 agent에게는 14를, 늦게 성공한 agent에게는 9의 Click Reward를 차등 부여함으로써 두 agent가 모두 기

본적으로 클릭을 성공하는 것을 목표로 하되, 상대 agent보다 늦게 클릭에 성공한 경우 reward를 일부 차감시켜서 부여하는 방식으로 설정했다. 초기에는 늦게 성공한 agent를 클릭 실패로 여겨서 -1의 click reward를 부여하였더니, 초반에 둘 중 한 agent가 우연히 여러 번 클릭에 먼저 성공할 경우 다른 agent는 클릭을 목표로 하지 않고 motor effort 만을 줄이는 방향으로 local minimum에 수렴하는 양상을 보였다. 따라서 이를 방지하기 위해 두 agent 모두 기본적으로 클릭을 성공하는 것을 목표로 하기 위해 뒤늦게 클릭에 성공하는 agent에게도 소량의 agent를 부여하도록 했다.

③ Episode Termination

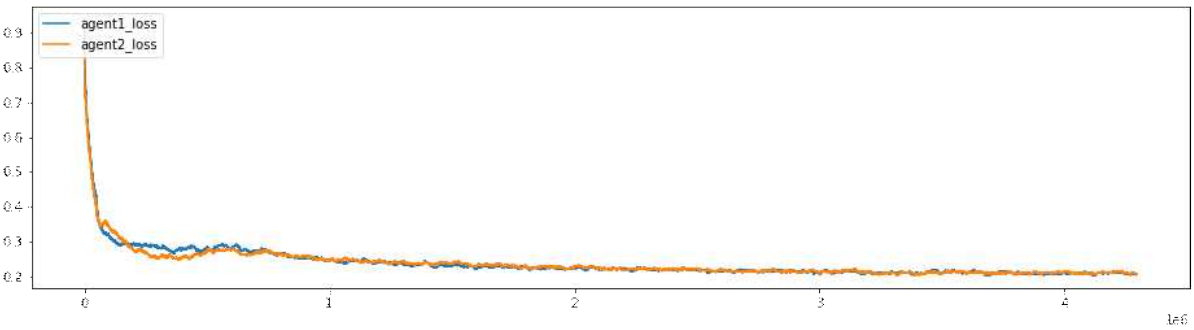
Fig 9. 적대적 강화학습 시 Episode Termination 기준

각 학습 Episode를 종료시키는 기준은 각 Agent에게 모두 Click 시도 기회를 1번씩 부여한 후 그 기회들이 모두 소진되었을 때 종료되도록 설정했다. 즉, 두 Agent의 클릭 '성공' 여부에 관계없이 각각 한번씩 클릭을 시도하면 해당 학습 Episode가 종료되고 다음 학습 Episode로 넘어가도록 설정했다. 단, 먼저 클릭에 시도한 agent가 있을 경우, 해당 agent에게는 먼저 클릭하는데 걸린 시간까지만 축적된 motor effort를 부여했다.

5. 연구 결과 및 분석

(1) 동일한 Agent 간의 적대적 강화학습 결과

① 학습 결과



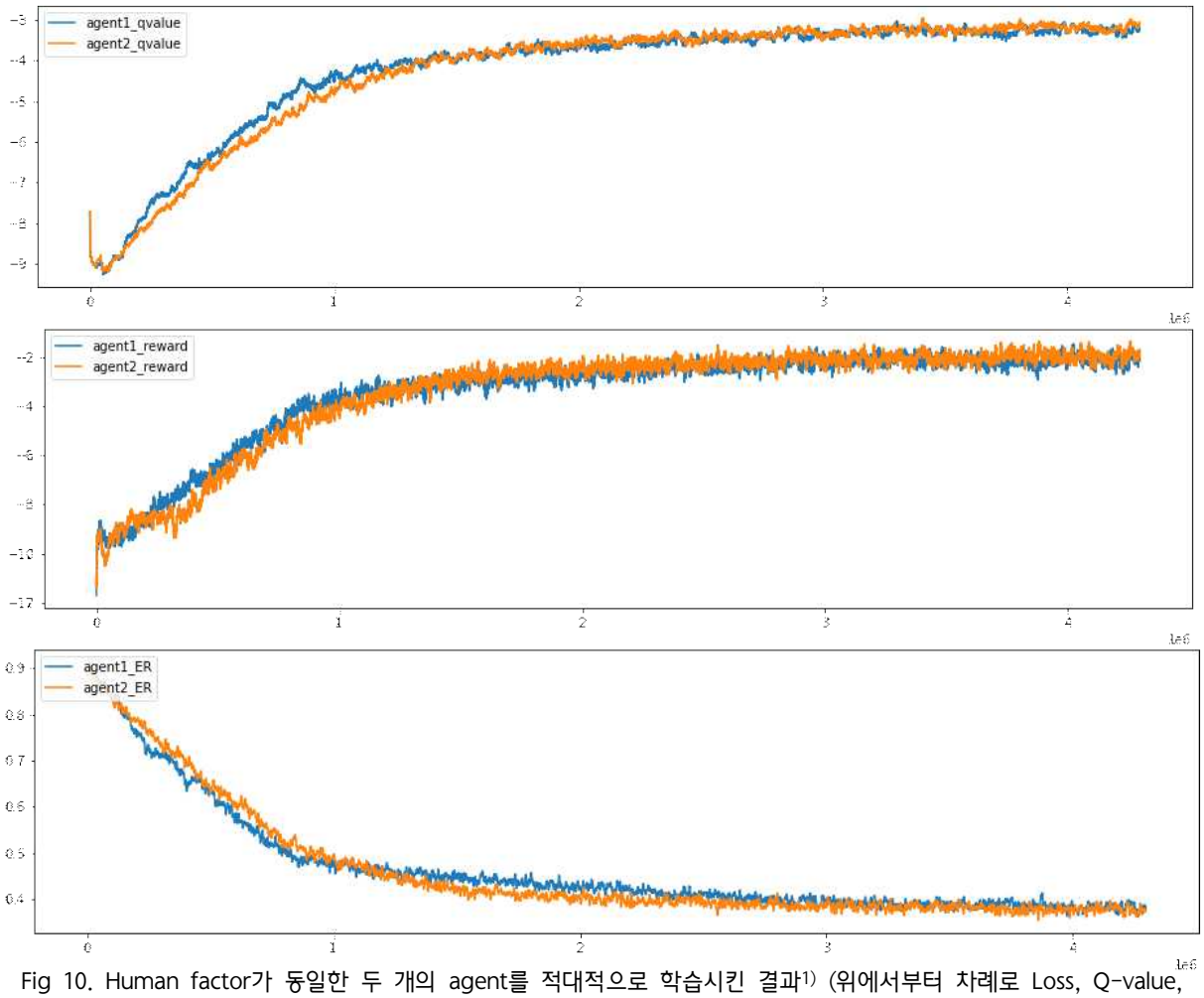


Fig 10. Human factor가 동일한 두 개의 agent를 적대적으로 학습시킨 결과[1] (위에서부터 차례로 Loss, Q-value, Reward, Click Failure rate)

Human Factor 값이 서로 동등한 2개의 agent[2]를 적대적으로 학습시킨 결과, Fig 10과 같은 학습 양상을 보였다. Loss는 단조 감소, Q-value와 Reward는 단조 증가하는 바람직한 양상임을 알 수 있고 그래프 전반적으로 비슷한 양상으로 학습이 잘 진행되어 둘 다 비슷한 수준으로 수렴했다. 초반에는 agent1이 agent2보다 먼저 학습이 잘 되어서 agent2를 이기다가 agent2도 학습이 잘 되어서 이후엔 agent1을 이기는 경우도 있음을 확인할 수 있다. 약 4백만 episode에 이르렀을 때에는 결국 두 agent 모두 타겟을 정확하고 빠르게 클릭하도록 학습이 잘 진행되어 click failure rate도 0.4 정도로 수렴했다. 또한, 앞서 기술한 reward 산정방식 덕분에 초반에 어느 한 agent가 우연히 먼저 타겟 클릭에 성공을 잘 할 경우, 다른 agent가 클릭하지 못해 local minimum에 빠지는 것을 방지할 수 있었다.

1) 2주간 약 4.3백만 개의 episode를 학습한 결과. 그래프는 5000개씩 moving average를 적용한 결과.
2) 기존 연구[1]에서 사용한 Human Factor 값과도 동일한 값 사용.

② 정성적 분석

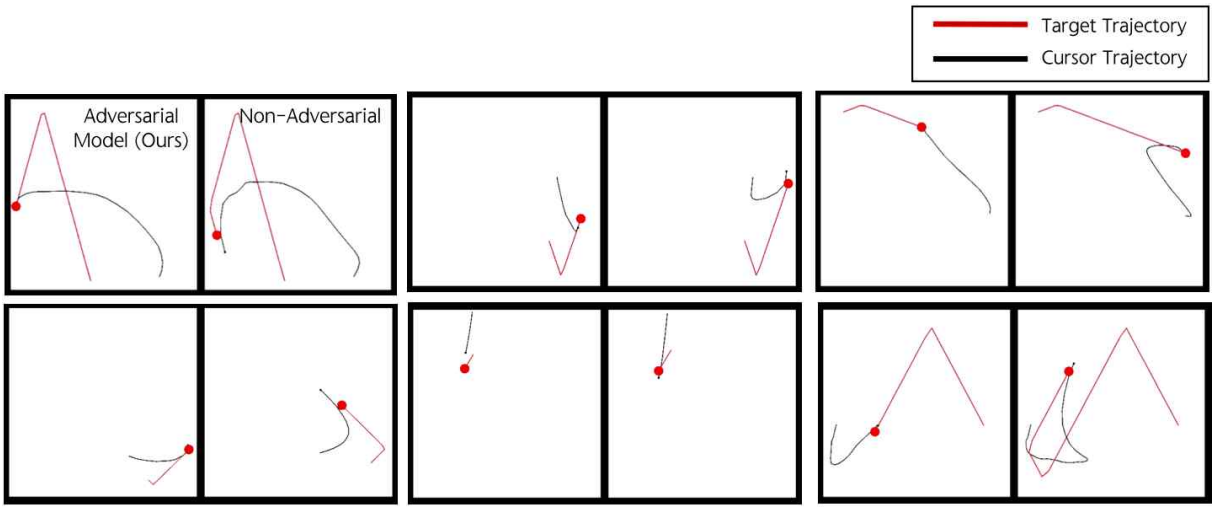


Fig 11. Human factor가 동일한 두 개의 agent를 적대적으로 학습시킨 결과 – 타겟 클릭 과정 시각화

본 연구에서 적대적으로 학습시킨 2개의 agent 중 하나의 agent를 선택하여[3] 적대적인 agent가 존재하지 않는 single agent 환경에서 학습시킨 기존의 agent (이하 기존 모델) [1]를 각각 1만개의 test episode에 test함으로써 타겟을 클릭하는데 까지의 커서의 이동경로(trajectory)를 비교해보았다. Fig 11를 보면, 타겟이 커서 위치에서 멀리 떨어진 초반부에는 두 agent의 이동 양상이 거의 동일했다. 하지만 타겟에 가까워지면서 적대적으로 학습시킨 모델은 기존 모델에 비해 보다 더 짧은 미래까지만을 예측하여 빨리 클릭을 시도하려는 경향이 있고, 기존 모델은 더 먼 미래까지를 고려해서 타겟과 자신의 이동경로를 비슷하게 따라가면서 클릭을 시도하는 경향이 있음을 알 수 있다. 결과적으로, 적대적인 환경에서 학습된 모델은 기존 모델의 policy에서 더 빠르게 클릭을 시도하는 방식으로 달라졌음을 확인할 수 있다.

③ 정량적 분석

본 연구에서는 기존의 SOTA 연구(참고 논문[1]) 에서 강화학습을 통해 학습된 Point and Click Model 의 Task 수행 결과를 정량적으로 분석하기 위해 채택한 방법 중 하나인 상관관계 분석을 이용하였다.

학습된 두 모델이 약 2만 번의 episode를 수행하며 측정된 데이터를 바탕으로, 각 Agent가 주어진 target의 radius 와 speed 가 변화함에 따라, 두 Agent 간의 Click Failure Rate 과 Trial Completion Time 이 어떠한 상관관계를 가지는지를 분석하였다. 이 때, target radius는 0.009 ~ 0.024 범위를 0.001 range로, target speed는 0 ~ 0.5s 범위를 0.05s range 로 binning 하였고, Click Failure Rate과 Trial Completion Time은 0.0 ~ 1.2 범위로 scaling 하여 그래프에 나타내었다.

3) 적대적으로 학습시킨 두 agent는 검증 결과 거의 동일한 모델 (다음 page 참조)

- Trial Completion Time & Click Failure Rate 비교

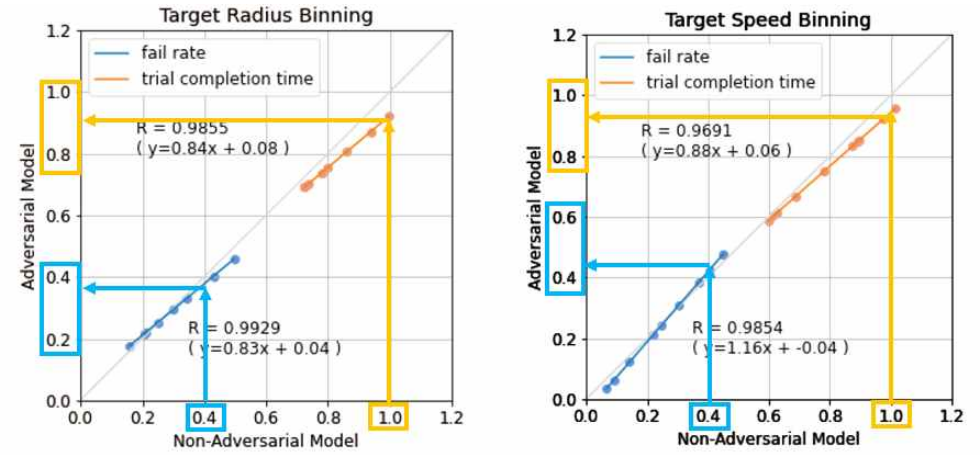


Fig 12. 기존 모델과 적대적으로 학습시킨 모델의 상관관계 분석결과

기존 SOTA model 과 적대적 환경에서 학습한 model 의 상관관계는 위의 그래프와 같이 Target의 radius가 작아질수록 그리고 Target의 speed는 커질수록 Click Failure Rate이 높아지고, Trial Completion Time이 증가하는 양의 상관관계를 나타내었다. 이는 적대적 모델이 기존의 SOTA Model 과 동일한 Task 수행 능력을 학습하였음을 의미하며, 또한 Click Failure Rate에 관한 상관 직선이 y=x 에 가깝게 나타난 것을 통해 적대적 환경에서 학습했음에도 불구하고 SOTA model 의 Task 수행 능력과 거의 차이가 나지 않음을 알 수 있다. 주목할 만한 것은 Trial Completion Time에 관한 직선이 모든 조건에서 y=x 아래에 위치하는데, 이는 적대적 환경에서 학습된 Agent가 SOTA model 보다 좀 더 급하게 Click을 시도한다는 것을 시사한다. 즉, 적대적으로 학습시킨 모델이 기존 모델에 비해 Trial Completion Time이 짧은 상태로 최적화되었음 확인할 수 있다.

다만, 위의 결과는 적대적으로 학습시킨 두 개의 모델 중 한 개의 모델을 임의로 선택하여 기존 모델과 비교한 것이었다. 따라서, 기존 모델과 적대적으로 학습시킨 모델을 위와 같이 비교하기 위해서는 적대적으로 학습시킨 2개의 모델이 서로 동일하다는 것을 보여야 한다.

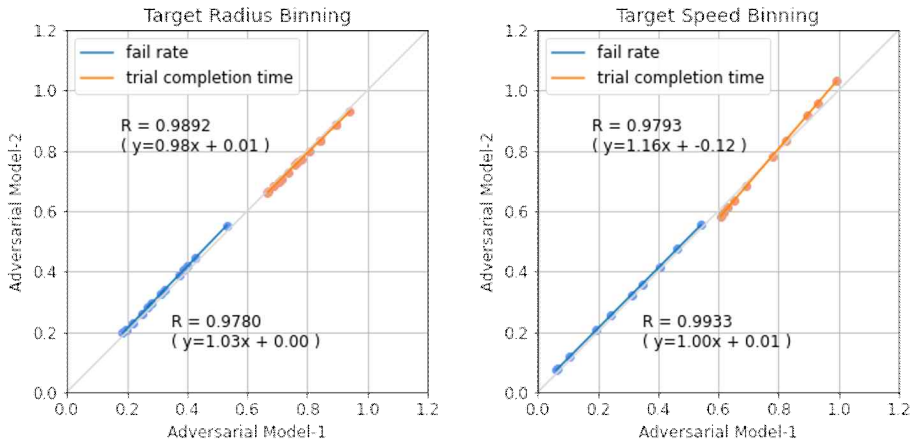


Fig 13. 적대적으로 학습시킨 2개의 모델 사이의 상관관계 분석

동일한 신체적, 인지적 능력을 가진 Agent 간의 Task 수행 결과 역시 위에서 나타난 것처럼 Target의 radius가 작아질수록, Target의 speed는 커질수록 Click Fail Rate이 높아지고, Trial Completion Time이 증가하는 양의 상관관계를 나타내었다. 또한 앞서 보았듯이 이는 기존의 SOTA Model 의 Test 결과와 같은 경향성을 나타냄을 알 수 있다. 경향성 뿐만 아니라 수치적인 측면에서도 거의 동일한 값을 나타났는데, 그래프에서 보이는 직선이 y=x 에 가깝게 나타난 것을 통해 어느 한 Agent의 학습이 일방적으로 잘 이루어진 것이 아니라 두 Agent 모두 비슷한 수준의 Task 수행 능력을 적대적 환경에서 학습할 수 있었음을 보여준다.

- **Mean Prediction Horizon 비교**

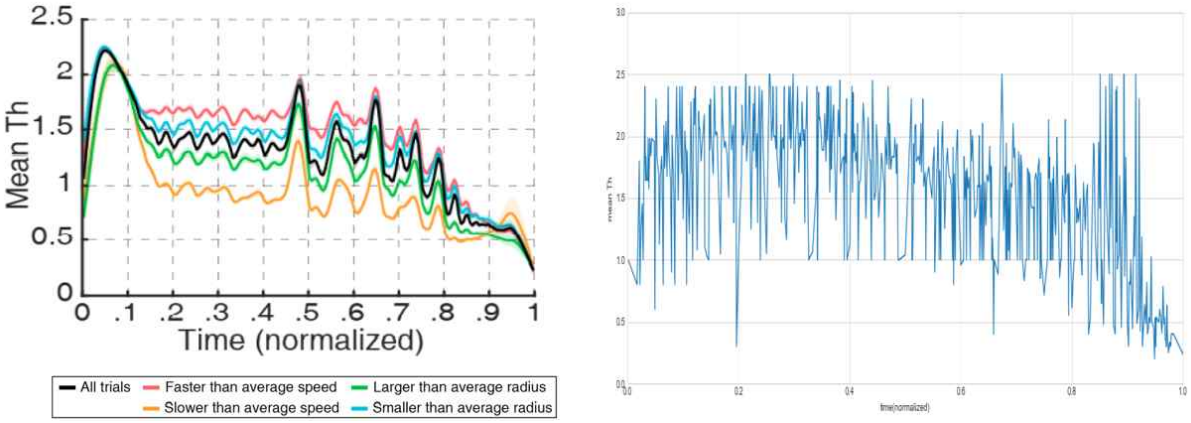


Fig 14. Human factor가 동일한 두 개의 agent를 적대적으로 학습시킨 결과 – mean prediction horizon 비교
(왼쪽: 기존 모델, 오른쪽: Ours)

Fig 14의 왼쪽 그래프는 기존 모델을, 오른쪽 그래프는 적대적으로 학습시킨 모델을 각각 약 2만개의 test episode에 대해 test하면서 Mean Prediction Horizon의 변화를 측정한 그래프이다. 왼쪽 그래프는 기존 연구 [1] 에서 진행한 실험 결과이기 때문에 두 그래프는 서로 동일한 episode 들에 대해 test한 결과는 아니지만, 통계적으로 유의미한 모수에 대한 각 agent의

Prediction horizon 변화를 나타내기 때문에 각 모델의 policy가 서로 상이하다는 것은 명백하다. 기존 모델은 episode 초반에 약 2초 정도까지 평균 prediction horizon이 증가한 후 1.5초 정도로 빠르게 줄어들고, 이후 클릭 시점까지 0.5초 부근으로 천천히 감소한다. 초반에 prediction horizon이 증가하는 이유는 타겟이 벽에 부딪치는 것을 이용하여 motor effort를 줄이는 방향으로 커서를 움직임으로써 더 높은 reward를 얻도록 최적화되었기 때문이다. 이후 클릭 이전에 prediction horizon이 감소하는 것은 지속적으로 이동하는 타겟을 클릭하기 위한 필수적인 행동이다. 반면, 적대적으로 학습시킨 agent는 클릭 직전에는 기존 모델과 비슷하게 prediction horizon이 감소하는 양상을 보였으나, episode 초반부터 중후반까지는 agent가 마치 경쟁자를 의식하는 것처럼 prediction horizon 값이 지속적으로 위아래로 변동하면서 타겟을 클릭하려고 하는 양상을 볼 수 있다.

(2) 서로 다른 Agent 간의 적대적 강화학습 결과

① 학습 결과

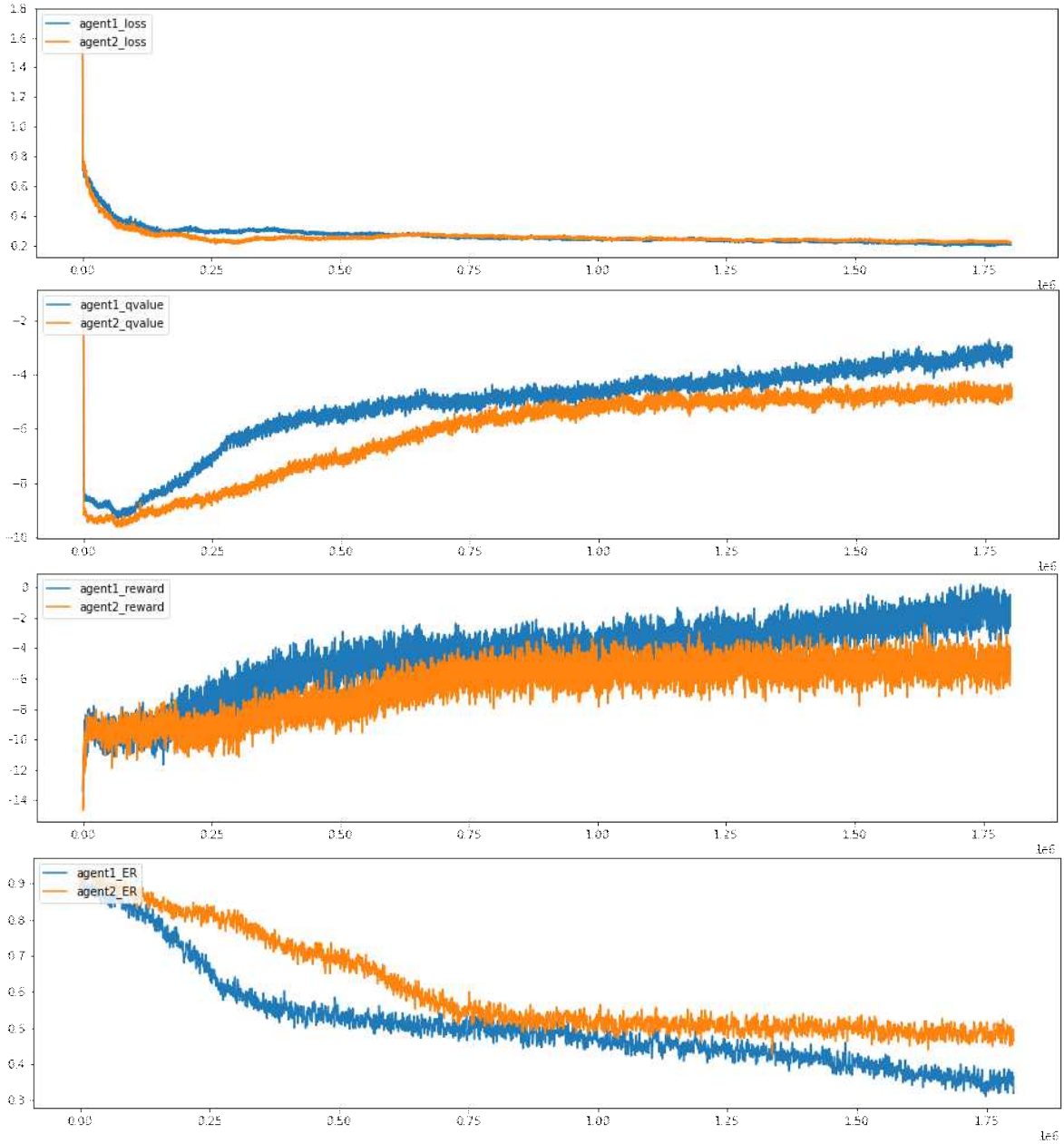


Fig 15. Human factor가 서로 다른 두 개의 agent를 적대적으로 학습시킨 결과[4] (위에서부터 차례로 Loss, Q-value, Reward, Click Failure rate)

Human Factor 값을 서로 다르게 변화시킨 두 개의 agent를 적대적으로 학습시킨 결과는 앞서 기술한 "5-(1) 동일한 Agent 간의 적대적 강화학습 결과"와 확연히 다른 모습을 확인할 수 있다 (Fig 15). Agent 1 (파랑색)은 Decision-Making Skill을 향상시키도록 Human Factor 값

4) Agent1: Improved decision-making skill agent(파랑색), Agent2: Improved motor execution skill agent(주황색). 1주간 약 1.8백만 개의 episode를 학습한 결과. 그래프는 500개씩 moving average를 적용한 결과.

을 조정한 모델이고 Agent 2 (주황색)는 Motor Execution 능력을 향상시키도록 Human Factor 값을 조정한 모델이다. 두 agent 모두 loss는 단조 감소, Q-value와 Reward는 단조 증가하는 바람직한 양상을 보였으나, Decision-Making Skill이 높은 agent가 reward를 더 빠르게 누적하는 양상을 보였고 click failure rate 또한 더 낮은 값으로 수렴되었음을 알 수 있다. 본 실험의 경우 시간 제약으로 인해 두 agent의 loss와 reward가 모두 수렴될 때까지 학습시키지는 못했으나, 전체적인 그래프의 양상으로 볼 때 평균적인 인간의 factor 중 Decision-Making Skill의 향상이 Motor Execution 능력의 향상보다 상대적으로 더 중요함을 알 수 있다.

② 정성적 분석

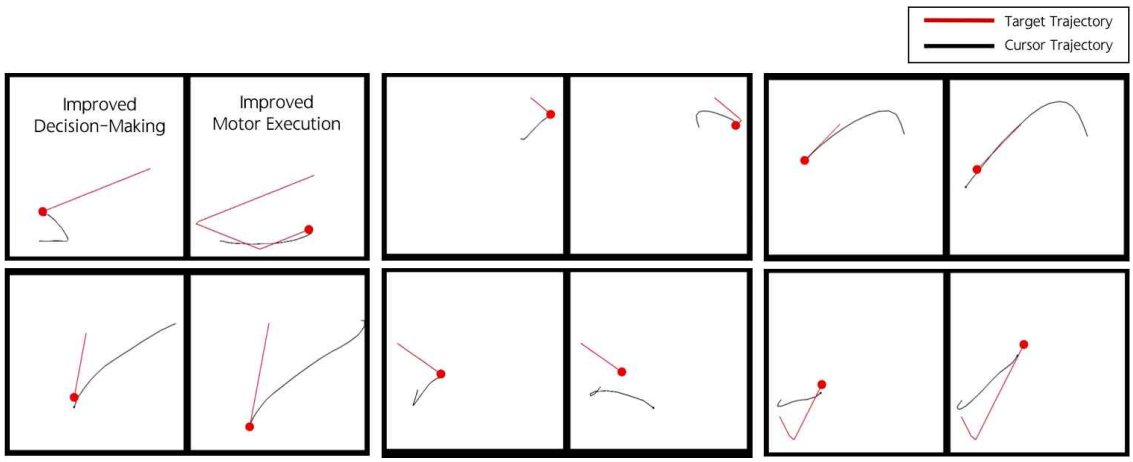


Fig 16. Human factor가 서로 다른 두 개의 agent를 적대적으로 학습시킨 결과 – 타겟 클릭 과정 시각화

앞서 학습시킨 두 agent를 마찬가지로 1만개의 test episode에 대해 test하면서 커서의 이동경로를 비교한 결과, Decision-Making Skill을 향상 시킨 agent가 Motor Execution 능력을 향상시킨 agent에 비해 타겟의 현재 위치로 바로 직진하는 경향이 있었음을 알 수 있다. 그에 반해, Motor Execution 능력을 향상 시킨 agent는 보다 먼 미래까지 고려해서 타겟에 대한 자신의 상대속도를 0으로 맞추려고 하고, 상대적으로 타겟에 곡선으로 접근하는 양상을 보였다.

③ 정량적 분석

- Trial Completion Time & Click Failure Rate 비교

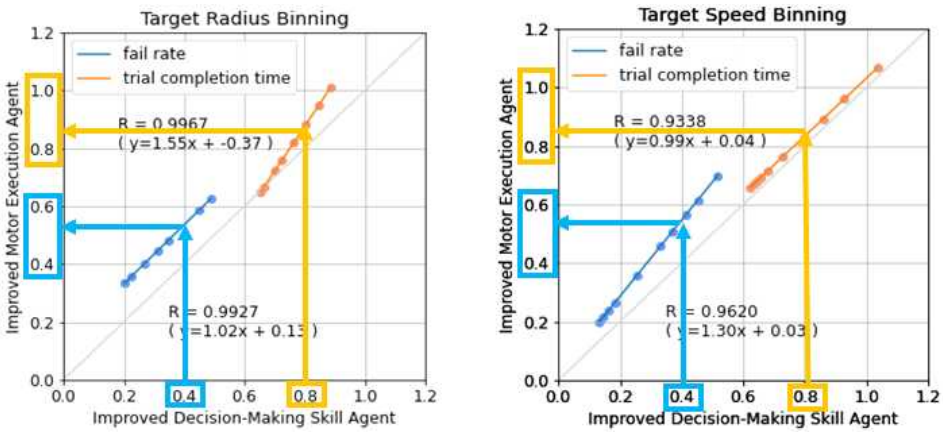


Fig 17. Human Factor 서로 다른 두 모델의 상관분석 결과

5-(1)에서처럼 Human Factor 가 서로 다른 두 모델에 대해서도 Trial Completion Time과 Click Failure Rate를 비교해보았다. 그 결과, Target Radius Binning과 Target Speed Binning에서 모두 Improved Decision Making Skill을 높였던 Agent가 Trial Completion Time이 더 짧고 Click Failure Rate 또한 더 작은 결과가 나왔다 (Fig 17). Trial Completion Time의 경우 약 0.05초, Click Failure Rate의 경우 약 0.15 정도로 Improved Decision Making Skill을 높였던 agent가 더 작은 값을 보였다. 이를 통해 Improved Decision Making Skill을 높였던 agent가 빠르면서도 더 정확하게 클릭에 성공한다는 것을 알 수 있고, 더 나아가 Point-and-Click Task를 수행할 때에는 Motor Execution 능력보다 Decision-Making Skill이 더 큰 영향을 미친다는 것을 알 수 있다.

6. 결론

본 연구의 최종 결론은 다음과 같이 크게 3가지로 정리할 수 있다.

(1) 적대적으로 학습시킨 모델 VS 적대적으로 학습시키지 않은 모델

먼저, 적대적으로 학습시킨 모델은 적대적으로 학습시키지 않았던 기존 모델 [1] 과 정성적, 정량적인 측면에서 Optimal Policy에 차이점이 존재했다. Target이 커서와 멀리 있을 때, 두 모델의 초기 Policy는 서로 비슷했지만, Target과 가까워졌을 때 적대적으로 학습시킨 모델은 근시안적인 시야(작은 Prediction Horizon 값)로 클릭 시도를 빨리하는 Policy를 보였으며, 그에 따라 Trial Completion Time을 더욱 낮추는 방향으로 최적화가 되었다. 반면, 적대적으로 학습시키지 않은 모델은 더 먼 미래까지를 고려하여 Target의 이동 경로를 예측한 후 클릭을 시도하는 Policy를 보였다.

이는 실제 세계의 적대적인 환경에서 reward를 최대화하는 Point-and-Click Task의 optimal policy가 무엇인지, 즉 실제로 적대적인 환경에서 Point-and-Click Task를 어떻게 수행하는 것이 가장 바람직한지에 대한 가이드라인을 제시한다. 실제 세계의 적대적인 환경에서 다른 유저와 Point-and-Click을 경쟁할 때는 먼 미래까지를 고려해서 복잡한 의사결정을 내리는 것보다 Target의 짧은 미래까지만을 예측해서 클릭을 시도하는 것이 최적의 전략이라는 것을 시사한다.

(2) Point-and-Click Task 수행시 Decision-Making Skill의 중요성을 강화학습으로 증명

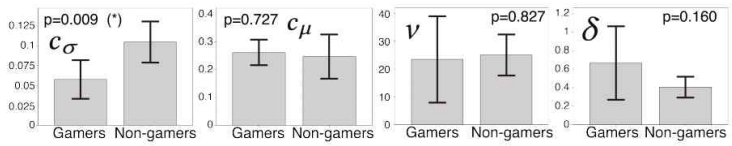


Figure 9. Comparison of four free parameters between the gamer group and the non-gamer group

Fig XX. 프로게이머와 일반인의 Decision-Making Skill 차이 (Park et al. [2])

두 번째로 본 연구는 Point-and-Click Task를 수행할 때에는 Decision-Making Skill이 Motor Execution보다 더 중요하다는 것을 강화학습으로 증명했다. 실제로 Point-and-Click Task 수행 능력이 뛰어난 FPS 게임 프로게이머와 일반인은 Decision Making Skill의 측면에서 두드러진 차이가 존재한다는 것을 보인 연구 [2] 가 있었는데, 본 연구에서는 이 사실을 강화학습으로 다시 한 번 증명해냈다.

(3) 활용 분야

앞서 기술한 두 가지 결론들을 활용한다면, 실제 세계의 적대적인 환경에서 Point-and-Click Task를 수행하는 사람들에게 유용한 가이드라인을 제공할 수 있을 것이라고 생각한다. 예를 들면 적대적인 환경에서의 Point-and-Click Performance를 최대화해야 하는 프로게이머들이 어떻게 하면 Point-and-Click Behavior를 최적화, 즉, 어떻게 하면 자신의 실력을 향상시킬 수 있을지에 대한 수학적인 가이드라인을 제공할 것이다. 또한, 그 외에도 인간과 Point-and-Click을 겨루는 Agent를 개발하는 등 E-sports 분야에서 폭넓게 활용될 수 있을 것이라고 생각한다.

7. 참고문헌

[1] Seungwon Do, Minsuk Chang, Byungjoo Lee, 2021. A Simulation Model of Intermittently Controlled Point-and-Click Behaviour. *CHI* (2021)

[2] Eunji Park, Byungjoo Lee, 2020. An Intermittent Click Planning Model. *CHI* (2020)