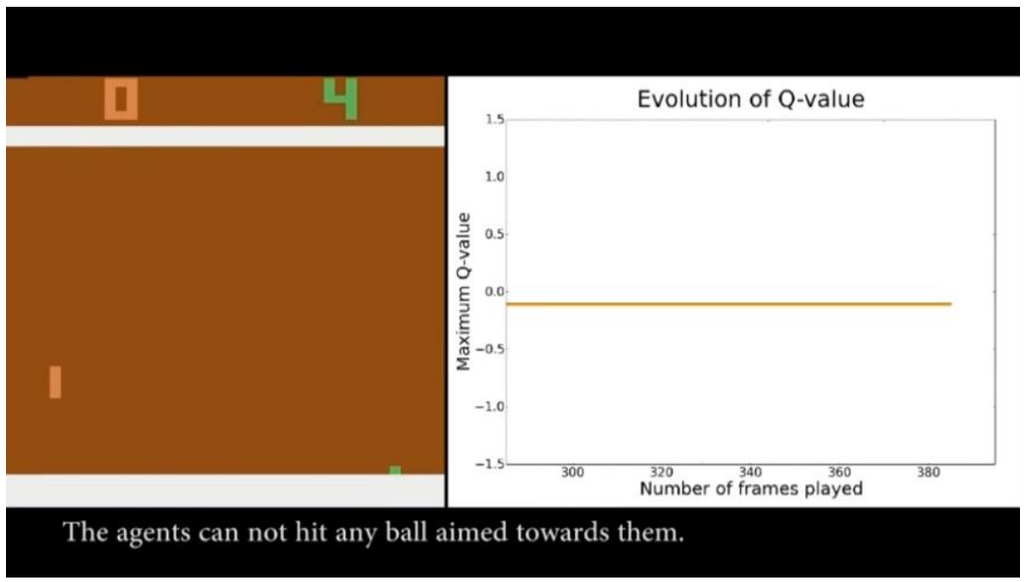
MADRL

Multiagent Cooperative and Competition with Deep Reinforcement Learning



김예찬(Paul Kim)

모두의연구소

https://github.com/NeuroCSUT/DeepMind-Atari-Deep-Q-Learner-2Player

Index

Abstract

Abstract

DeepMind가 제안한 **Deep Q-Learning Network**을 사용해 Multiagent system으로 확장하고 Pong환경 속에서 **IQN(Independent Q-Networks)**로 두 개의 agent들이 control되는지를 조사함

Pong환경의 **Reward스키마를 사전에 정의**해서 **competitive**한 경우와 **cooperative**한 행위가 어떻게 나타나는지를 확인

Competitive한 agents는 가정한 경우에는 score를 올리게 끔 학습이 되는 것을 확인

Cooperative한 agents를 가정한 경우에는 두 개의 에이전트가 가능한 서로 오랫동안 공을 keeping하는 것을 확인

Competitive에서 Cooperative의 진행을 설명(Reward의 값을 순차적으로 변경시켜서 확인)

결과적으로 이 연구논문의 저자들은 Deep Q-Network를 복잡한 환경 속에서 다중 에이전트 시스템의 decentralized(분산) 학습을 연구하기 위한 실용적인 방식을 제안함

Introduction

강화학습의 철학

Biological 혹은 Engineered agents는 예측 불가능성에 대처해야 하며 trial-and-error를 통해서 새롭고 변화되는 환경에서 행동을 적응시킬 수 있음

강화학습에서 에이전트는 환경과 상호작용하면서 수집한 보상에 따라 행동을 수정하고 이 때 보상을 최대화하기 위해서 에이전트는 복잡한 장기전략을 구현하는 방식을 터득하게 됨

DQN과 Multi-Agent

가능한 시나리오들이 너무나도 많기 때문에 강화학습은 대게 단순한 환경에만 국한되었고, 환경의 dynamics에 대한 추가 정보로 도움을 받아서 접근했었음

하지만 DeepMind가 Video게임과 같은 고차원적이고 복잡한 환경에 RL을 적용해 성과를 냈고, super-human의 성능을 달성. 기억할 점은 raw sensory input(이미지)와 reward signal(게임점수)만을 사용

Deep Q-Network(Convolution Neural Network를 Q-Learning으로 representation하는 모델)를 사용했고 당시(2015)를 기준으로 Model-Free 방식으로 State-of-the-art를 달성

다른 게임들을 배우기 위해서 동일한 알고리즘이 사용한다는 것은 general한 application에 대한 가능성을 시사함

Introduction

저자들의 연구방향

강화학습의 철학

DQN을 사용하고 raw screen image와 reward signal을 input으로 받는 환경으로 **Atari**를 사용함. 그리고 두 에이전트가 **정의된 rewarding scheme**로 training될 때 복잡한 환경 속에서 어떻게 행동하고 상호작용하는지를 확인하고자 함

DQN과 Multi-Agent

Competitive한 에이전트는 **점수를 올리는 것**을 학습하지만 **Cooperative**한 경우에는 **공을 최대한 유지하는 전략**을 찾아내려고 함.

Competitive 모드와 Cooperative모드 사이의 **중간 상태를 연구**하고 Competitive에서 Cooperative로의 진행을 관측하기 위해서 **rewarding schemes를 tuning**

다른 게임들을 배우기 위해서 동일한 알고리즘이 사용한다는 것은 general한 application에 대한 가능성을 시사함

Method : DQN

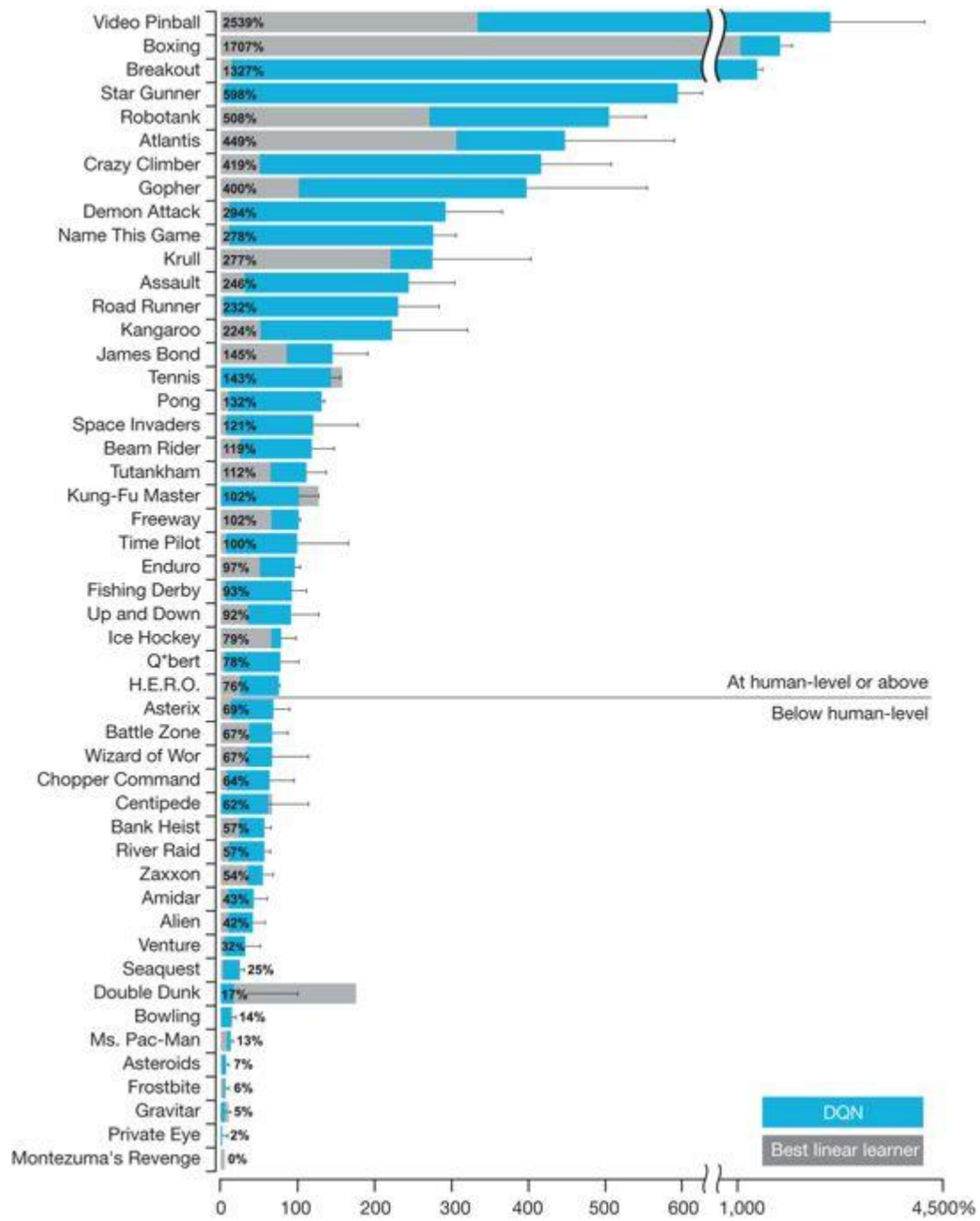
Q-Learning과 DQN

RL의 목표는 dynamic한 환경에서 에이전트의 accumulative long term reward를 maximize하는 정책을 찾는 것.

Agent가 환경의 dynamics나 reward와 같은 implicit한 정보가 없을 때 문제가 됨. 이 때 Model-Free방식인 Q-Learning을 당시(약 2015년 전후)에는 많이 사용했음

Q-Learning은 환경의 특정한 상태에서 행동의 가치를 평가할 수 있음

DeepMind는 이러한 Q-Learning에 Convolution Neural Network를 활용해 Q-Value를 approximate해서 비디오 게임에서 super-human 성능을 달성함



Method : DQN

DQN과 Multi-Agent

두 개 이상의 에이전트가 환경을 공유하는 경우는 싱글 에이전트에 비해 이해가 쉽지 않음

학습의 분산된 특성은 새로운 이점을 제공하지만, 좋은 학습 목표의 정의나 알고리즘의 수렴 및 일관성 같은 점을 고려해야 함

Ex) Multi-Agent Setting의 경우 환경의 state transition와 reward가 모든 에이전트의 공동 작업에 영향을 받음. 에이전트1의 행동에 대한 가치가 다른 에이전트2의 행동에 따라 변경되기 때문에 각 에이전트는 다른 학습 에이전트를 추적해야만 함. 일반적으로 다른 에이전트가 있는 상태에서 학습을 수행하면 각 에이전트의 stability와 adaptive behavior사이의 균형이 필요함

다중 에이전트에 Q-Learning알고리즘을 적용하는 사례가 있지만 그 당시를 기준으로는 특정 유형에 따라서 제약이 존재

(그 당시 기준)Simplicity, decentrailized nature, computational speed 그리고 작업의 범위에 대해 일관된 결과를 산출할 수 있는 능력이 DQN에 있기 때문에 이 방법을 사용.

파라미터는 Human-level control through deep reinforcement learning 의 논문과 동일하게 설정

Method : Adaptation of the Code for the Multiplayer Paradigm

OpenAI Gym과 MultiAgent Setting

기존의 Human-level control through deep reinforcement learning 논문과 함께 공개된 코드는 멀티 에이전트 게임을 수행할 수 있는 환경을 공개하지 않았음

에이전트는 DQN으로 독립적으로 구현 되어있고, Atari 게임은 다중 플레이어를 허용하지만 에이전트와 에뮬레이터 간의 communication protocol은 단일 플레이어를 사용하도록 제한을 걸어놓음

게임 화면은 Fully Observable하고 두 에이전트 간에 공유되므로 동시에 여러 에이전트에 게임 상태를 제공하기 위해서 추가적인 수정이 필요하지 않았음

Method : Game Selection

Game환경 결정 기준

ALE(Atari Learning Environment)는 61개의 게임을 지원하고 있지만 2개의 플레이어 모드가 가능한 게임환경은 굉장히 적음. 그래서 실험을 위해서 아래의 **3가지**의 기준으로 실험환경을 필터링

1. Real-time two-player mode가 있어야함

: 예를 들어 Breakout의 경우는 두 명의 플레이어가 번갈아 수행해야하는 단점이 있으므로 사용 X

2. Deep Q-Learning 알고리즘이 싱글 플레이어 모드에서 인간 수준 이상의 게임을 수행한 것이 검증된 환경만 사용함

: 예를 들어 “Wor of Wizard”라는 게임은 2인 모드를 제공하지만 미로 탐색은 기존의 DQN으로 마스터할 수 없었음

3. 게임 자체가 경쟁 모드인 경우

: 그리고 reward function을 변경해서 Cooperation한 경우와 Competitive한 경우를 전환할 수 있는 게임이어야 함

결과적으로 Pong게임 환경이 모든 기준을 만족하기 때문에 선택하게 되었고 상대적으로 널리 알려진 게임이므로 선택!!

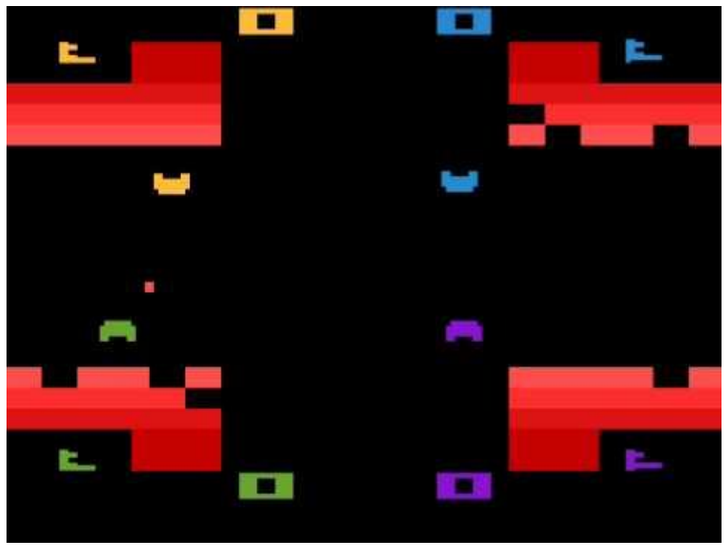
Method : Game Selection

Game환경 결정 기준

Pong에서 각 에이전트는 화면의 왼쪽과 오른쪽에 위치한 패들 중 하나에 해당
두 에이전트가 할 수 있는 행동의 조합은 **4가지** : **위로 이동**, **아래로 이동**, **정지** 및 **발사**(볼을 시작하거나 게임 시작)

어떤 행동이 결정되는지는 각각의 DQN에 의해 두 에이전트가 별개로 결정이 된다는 것을 기억해야 함
현재 연구 범위를 벗어나지만 경쟁과 협업에 관해 흥미로운 과학적 질문과 잘 어울리는 다른 두 가지 게임에 대해서 확인(Outlaw, Warlords)

1. Outlaw는 제한된 게임 모드에서 죄수의 딜레마에 대한 실시간 근사치로 볼 수 있는 간단한 슈팅게임
2. Warlords는 최대 4명의 플레이어가 적이 나타났을 때 collaboration이 나타나는지를 테스트할 수 있는 게임



Method : Rewarding Schemes

Rewarding Schemes

실험의 목표는 agent에게 보상을 받는 방법에 따라서 다양하게 조정된 행동이 나타나는 것을 연구하는 것임.
이러한 목표에 따라서 여러가지의 Rewarding Scheme들을 설정했음

1. Score More than the Opponent(Fully Competitive)

	Left player scores	Right player scores
Left player reward	$+1$	-1
Right player reward	-1	$+1$

2. Loosing the Ball Penalizes Both Players(Fully Cooperative)

	Left player scores	Right player scores
Left player reward	-1	-1
Right player reward	-1	-1

3. Transition Between Cooperation and Competition

	Left player scores	Right player scores
Left player reward	ρ	-1
Right player reward	-1	ρ

Score More than the Opponent(Fully Competitive)

Competitive한 Zero-Sum Schemes : 이기면 +1, 지면 -1

Pong에서는 공을 마지막으로 터치한 Player가 플러스 점수를 얻고, 공을 놓치는 Player가 마이너스 점수를 얻음.

기본적인 **Zero-Sum게임**이 되는 구조. 만약 왼쪽 플레이어가 긍정적인 보상을 얻게 되면 오른쪽에 있는 Player는 부정적인 보상을 받게 되고 반대도 마찬가지

이러한 경우를 **완전한 경쟁모드**라고 부름

	Left player scores	Right player scores
Left player reward	+1	−1
Right player reward	−1	+1

Loosing the Ball Penalizes Both Players(Fully Cooperative)

Cooperative Mode : Ball을 잃어버리면 페널티

가능한 에이전트들이 오랫동안 게임에서 공을 유지하는 법을 배우는 것.

공을 놓치는 경우에 두 Player에게 **부정적인 보상**을 설정함

기억할 점은 어느 Player가 공을 쳐서 상대방에게 던지는지는 중요하지 않고 **긍정적인 보상을 받을 수 있는 설정은 존재하지 않음**

tip) 또 다른 가능한 협력 모드는 각각의 나가는 공에서 두 선수에게 보상을 하는 것이지만 실험을 하지 않았다고 함

	Left player scores	Right player scores
Left player reward	−1	−1
Right player reward	−1	−1

Transition Between Cooperation and Competition

Cooperative한 경우와 Competitive한 경우 모두 공을 잃어버리면 penalty를 받음

두 가지의 전략을 구별하는 것은 Reward Matrix의 대각선 상에 존재하는 값임. 보상 에이전트들이 다른 Player를 지나쳐서 공을 던지면서 얻음. Reward의 가치가 -1에서 1로 점진적으로 변화도록 허용하면 Competitive와 Cooperation에 있는 중간 시나리오를 확인할 수 있음

포인트 /rho 를 점수로 하는 보상이 -1에서 1로 변화하고 공의 손실에 대한 페널티는 -1로 고정시켜서 접근함.

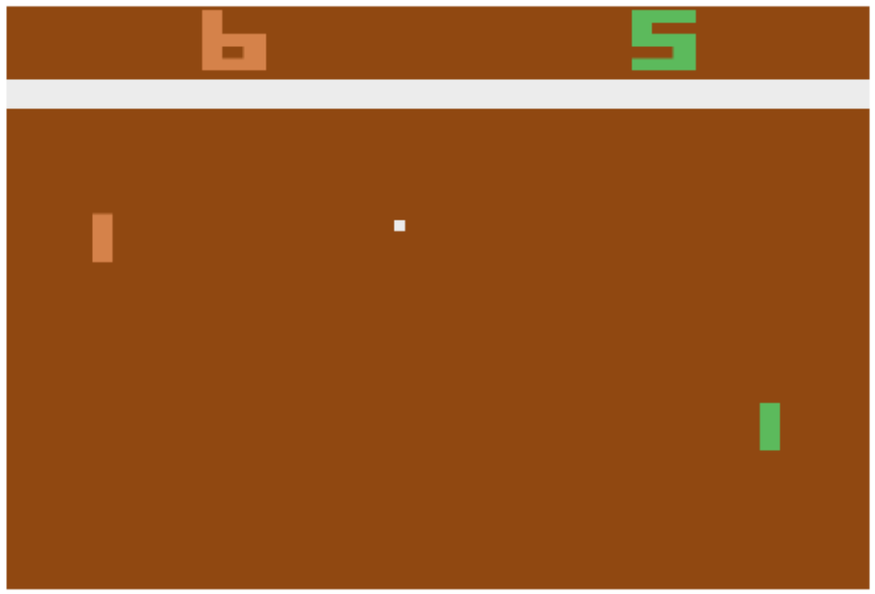
	Left player scores	Right player scores
Left player reward	ρ	-1
Right player reward	-1	ρ

Training Procedure

모든 실험에서, 50 Epochs를 설정하고 timestep은 1~250000을 설정

Frame skipping을 사용하기 때문에 에이전트가 4번째 프레임을 기준으로 수행됨. “visible frame”, “frame”, “time step”을 사용함

Epsilon greedy를 사용하여 Epsilon decay를 설정(1에서 0으로 0.05를 설정하여 접근)



Collecting the Game Statics

Pong환경에서 에이전트의 행동에 대한 정량적인 측정을 얻기 위해서 게임의 이벤트를 확인하고 계산했음(볼이 Paddle 또는 Wall에 부딪히는 경우)

- *stella : "A Multi-Platform Atari 2600 VCS Emulator" 를 사용해서 이벤트를 감지*

Cooperative와 Competitive를 양적으로 평가하기 위해서 모든 rewarding scheme에 대해서 각 훈련 epoch이 끝난 이후에 측정된 결과를 수집함. 각 epoch이 끝난 이후에 두 Player의 현재 state에 있는 Q-Network를 활용하며 random seed의 값을 변경해 10번의 게임을 실험하고 통계치를 수집함. Testing단계에서는 epsilon(exploration rate)의 값을 0.01로 설정함

1. Average paddle-bounces per point

point별 평균 paddle-bounces는 두 Player 중 한 명이 득점하기 전에 공이 두 선수 사이에서 몇 번 튀는지를 세어보는 것임. Random play를 하는 에이전트는 공을 거의 치지 못함. 실험에서는 Paddle-bounces라고 명명

2. Average wall-bounces per paddle-bounce

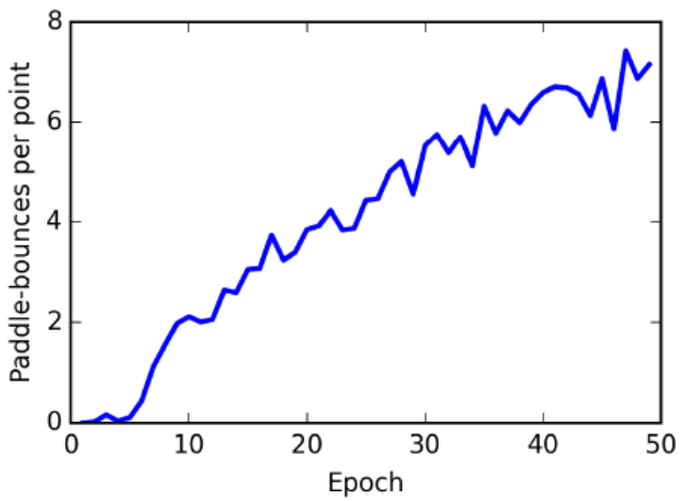
paddle-bounce당 평균 wall-bounce는 상대 Player에게 도달하기 전에 위쪽과 아래쪽 벽에서 얼마나 많이 공이 부딪히는지를 의미함. 상대방 Player에게 닿기 전에 여러 번 벽을 튕기도록 날카로운 각도로 공을 칠 수 있음. 실험에서는 wall-bounce라고 명명

3. Average serving time per point

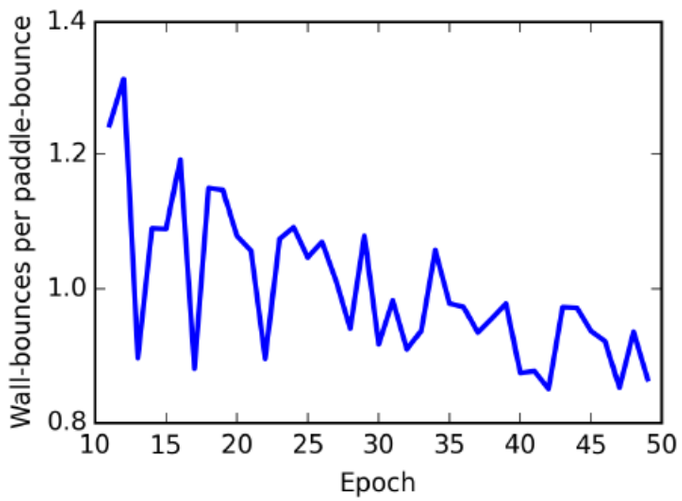
point별 평균 serving시간은 Player가 패한 후 경기를 다시 시작하는데 걸리는 시간. 게임을 다시 시작하려면 득점을 얻은 Player가 fire라는 특정 명령을 수행해야함. 실험에서는 serving time이라고 명명

Results : Emergence of Competitive Agents

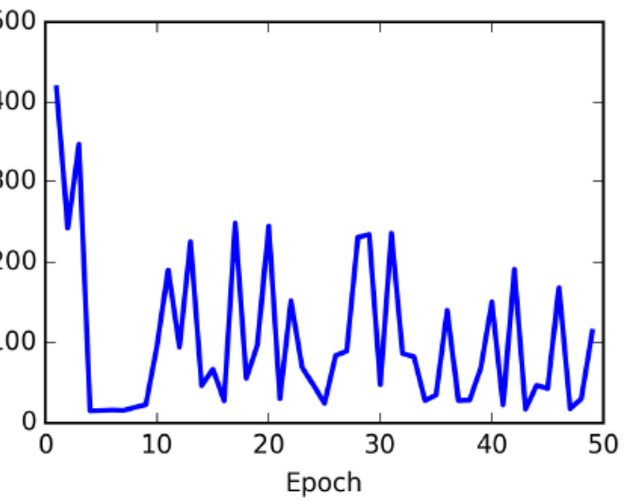
Results : Emergence of Competitive Agents



(a) Paddle-bounces per point



(b) Wall-bounces per paddle-bounce



(c) Serving time per point

Figure 2: Evolution of the behaviour of the competitive agents during training. (a) The number of paddle-bounces increases indicating that the players get better at catching the ball. (b) The frequency of the ball hitting the upper and lower walls decreases slowly with training. The first 10 epochs are omitted from the plot as very few paddle-bounces were made by the agents and the metric was very noisy. (c) Serving time decreases abruptly in early stages of training- the agents learn to put the ball back into play. Serving time is measured in frames.

Results : Emergence of Competitive Agents

Results : Emergence of Competitive Agents

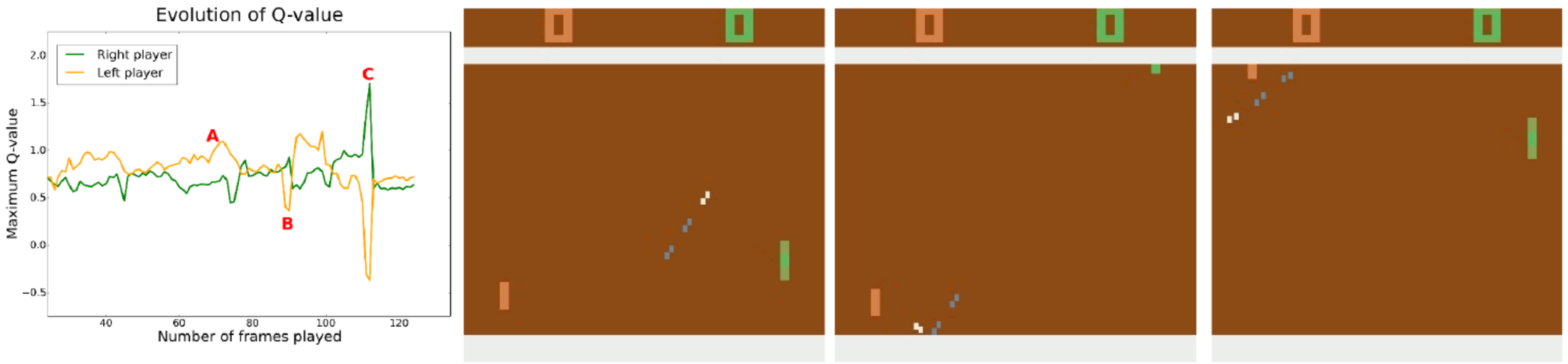
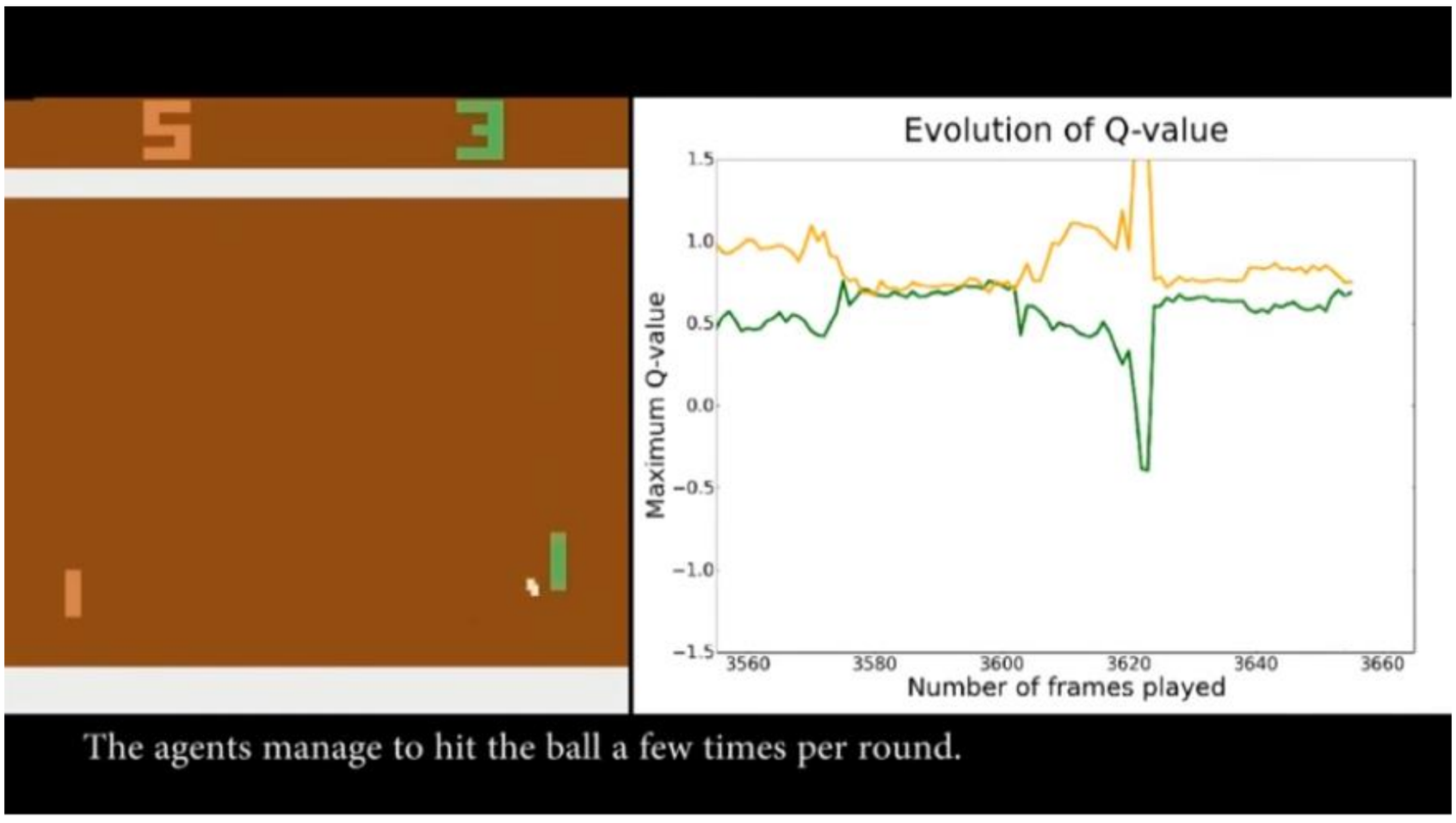


Figure 3: A competitive game - game situations and the Q-values predicted by the agents. A) The left player predicts that the right player will not reach the ball as it is rapidly moving upwards. B) A change in the direction of the ball causes the left player's reward expectation to drop. C) Players understand that the ball will inevitably go out of the play. See section 4.3 for videos illustrating other game situations and the corresponding agents' Q-values.

Results : Emergence of Competitive Agents

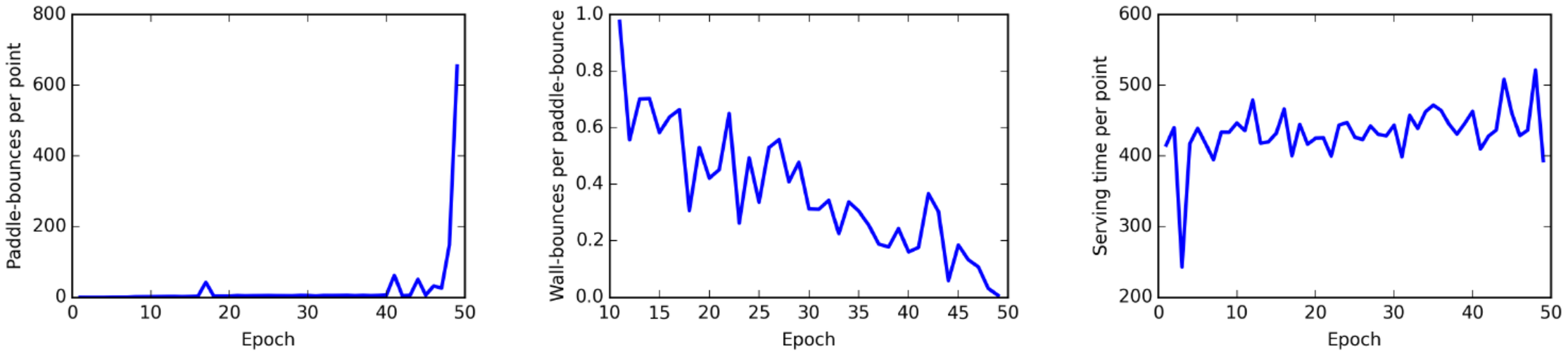
Competitive Agents결과 동영상



The agents manage to hit the ball a few times per round.

Results : Emergence of Cooperative Agents

Results : Emergence of Cooperative Agents



(a) Paddle-bounces per point

(b) Wall-bounces per paddle-bounce

(c) Serving time per point

Figure 4: Evolution of the behaviour of the collaborative agents during training. (a) The number of paddle-bounces increases as the players get better at reaching the ball. (b) The frequency of the ball hitting the upper and lower walls decreases significantly with training. The first 10 epochs are omitted from the plot as very few paddle-bounces were made by the agents and the metric was very noisy. (c) Serving time increases - the agents learn to postpone putting the ball into play. Serving time is measured in frames.

Results : Emergence of Cooperative Agents

Results : Emergence of Cooperative Agents

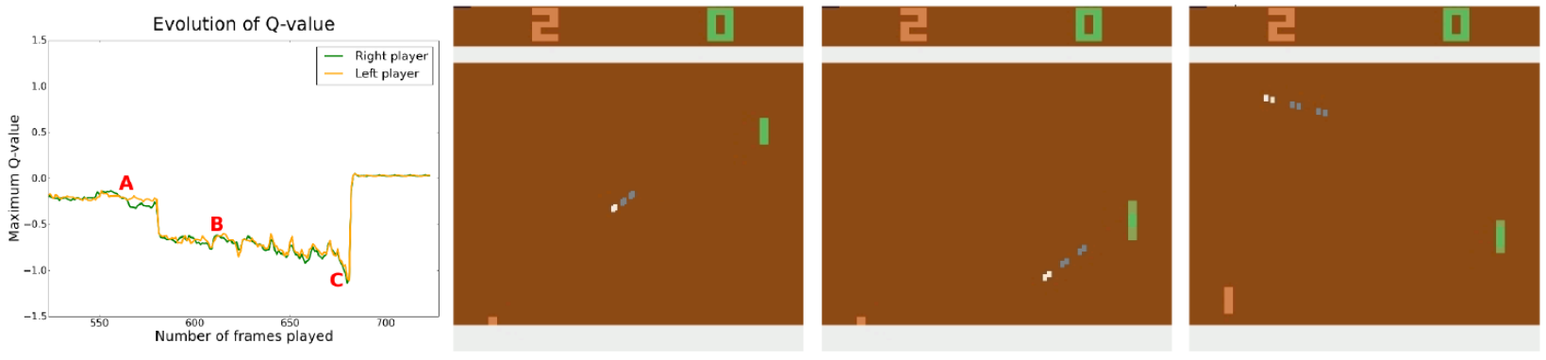


Figure 5: Cooperative game. A) The ball is moving slowly and the future reward expectation is not very low - the agents do not expect to miss the slow balls. B) The ball is moving faster and the reward expectation is much more negative - the agents expect to miss the ball in the near future. C) The ball is inevitably going out of play. Both agents' reward expectations drop accordingly. See section 4.3 for videos illustrating other game situations and the corresponding agents' Q-values.

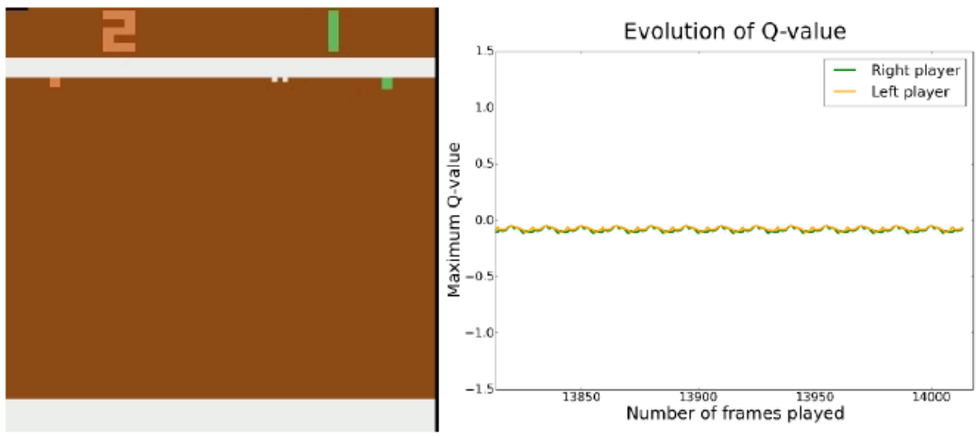
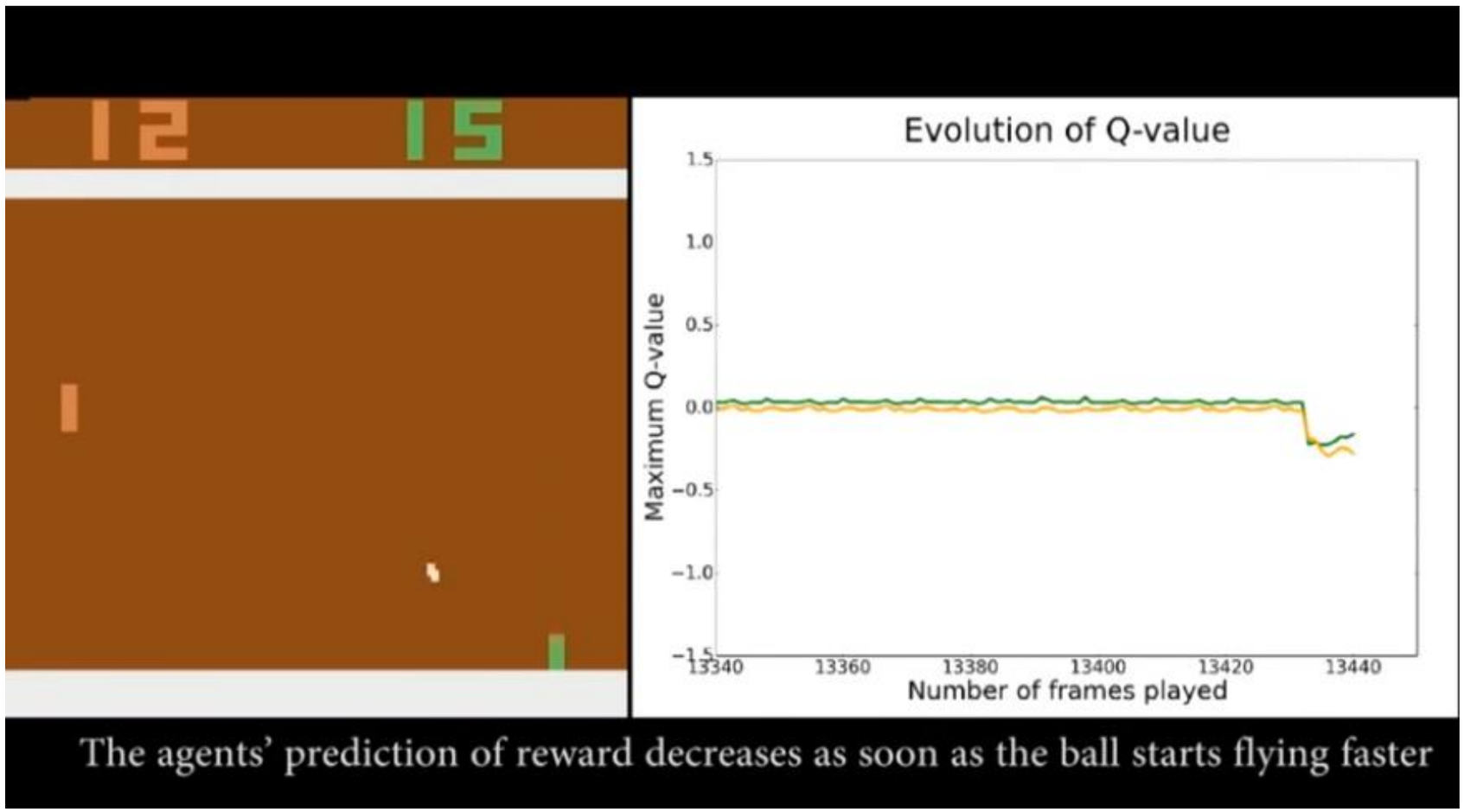


Figure 6: In the cooperative setting the agents sometimes reach a strategy that allows them to keep the ball in the game for a very long time.

Results : Emergence of Cooperative Agents

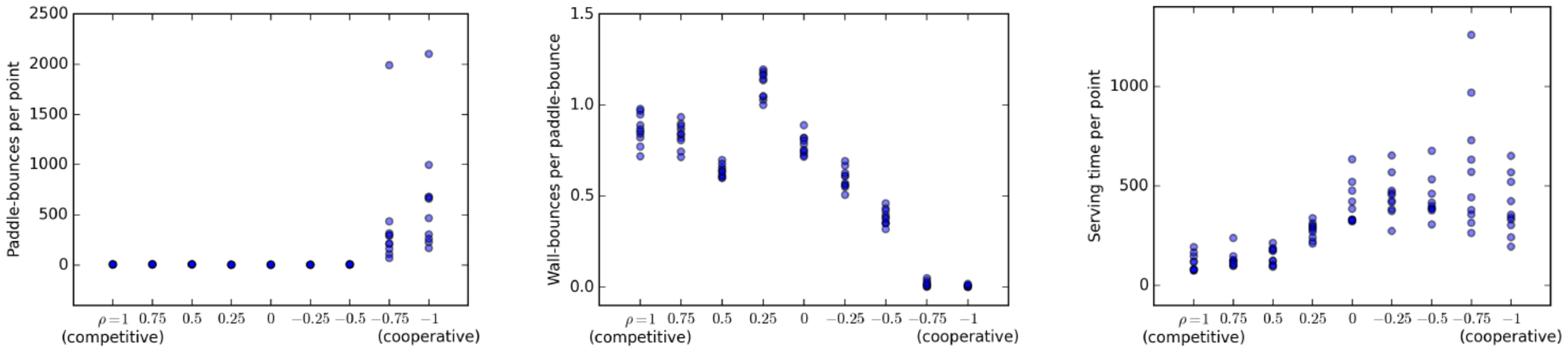
Cooperative Agents결과 동영상



The agents' prediction of reward decreases as soon as the ball starts flying faster

Results : Progression from Competition to Collaboration

Results : Progression from Competition to Collaborarion



(a) Paddle-bounces per point

(b) Wall-bounces per paddle-bounce

(c) Serving time per point

Figure 7: Progression of behavioural statistics when passing from competitive to collaborative rewarding schemes. (a) The number of touches increases when the agents have a strong incentive to collaborate. (b) Forcing the agents to collaborate decreases the amount of angled shots that bounce off the walls before reaching the opposite player. (c) Serving time is significantly shorter when agents receive positive rewards for scoring.

Results : Progression from Competition to Collaboration

Results : Progression from Competition to Collaborarion

Agent	Average paddle-bounces per point	Average wall-bounces per paddle-bounce	Average serving time per point
Competitive $\rho = 1$	7.15 ± 1.01	0.87 ± 0.08	113.87 ± 40.30
Transition $\rho = 0.75$	7.58 ± 0.71	0.83 ± 0.06	129.03 ± 38.81
Transition $\rho = 0.5$	6.93 ± 0.49	0.64 ± 0.03	147.69 ± 41.02
Transition $\rho = 0.25$	4.49 ± 0.43	1.11 ± 0.07	275.90 ± 38.69
Transition $\rho = 0$	4.31 ± 0.25	0.78 ± 0.05	407.64 ± 100.79
Transition $\rho = -0.25$	5.21 ± 0.36	0.60 ± 0.05	449.18 ± 99.53
Transition $\rho = -0.5$	6.20 ± 0.20	0.38 ± 0.04	433.39 ± 98.77
Transition $\rho = -0.75$	409.50 ± 535.24	0.02 ± 0.01	591.62 ± 302.15
Cooperative $\rho = -1$	654.66 ± 542.67	0.01 ± 0.00	393.34 ± 138.63

Table 4: Behavioural statistics of the agents as a function of their incentive to score.