カードゲーム型対戦環境への深層強化学習手法の適用

創発ソフトウェア研究グループ

B3　西村 昭賢

発表の流れ

- はじめに
- 要素技術
- 構築環境
- 実験
- 結果
- まとめと今後の課題

発表の流れ

- はじめに
- 要素技術
- 構築環境
- 実験
- 結果
- まとめと今後の課題

深層強化学習による問題解決

- 自動運転
- ロボットの制御
- 推薦システム

⇒ 様々な実世界の問題解決へ応用

ゲームへの応用

- 完全情報ゲーム (囲碁, 将棋) への応用

  ⇒ プロを圧倒

  AlphaGo (2015), AlphaZero (2017)

- 不完全情報ゲーム (ポーカー, 麻雀) への応用

  ⇒ 現在注目されている

本研究の目的

- カードゲーム型対戦環境の構築
- 構築環境への深層強化学習の適用
- 構築環境のゲームバランスの調整

本発表

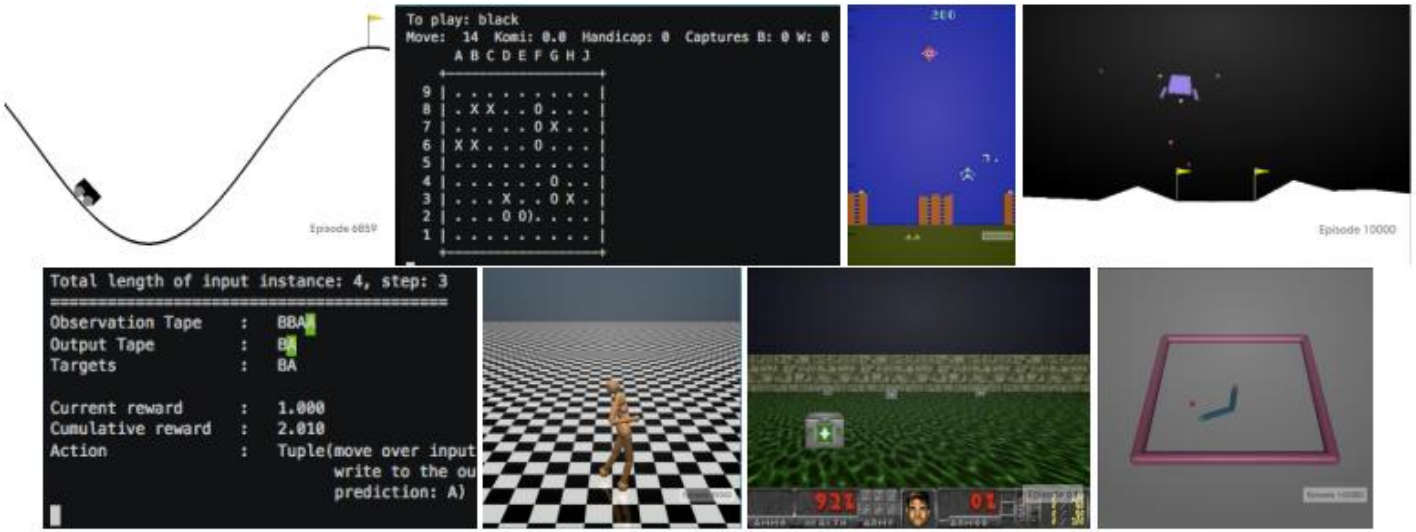
- カードゲーム型対戦環境の構築
- 構築環境への深層強化学習の適用
- モンテカルロ探索による強化学習の適用

発表の流れ

- はじめに
- 要素技術
- 構築環境
- 実験
- 結果
- まとめと今後の課題

OpenAI Gym

- OpenAI 社が提供する強化学習用シミュ レーションライブラリ
- 様々な学習環境が提供されている
- インターフェースを用いて自作環境作成



Brockman, G., OpenAI Gym, arXiv e-prints, 2016.

Q学習

- 代表的な価値ベースの強化学習手法の 1 つ

- Q 値を以下の式に従って 1 ステップごとに更新していく

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\{r_{t+1} + \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\}$$

TD誤差

α : 学習率 (Q 値の更新の度合い)

γ : 割引率 (将来の価値の割引度合い)

Deep Q Network (DQN)

- Q 学習では状態や行動の次元数が増えると現実的に計算ができなくなる

  ⇒ 深層学習を用いることで学習可能に

- Experience Replay や Fixed Target Network により安定した学習が可能になる

モンテカルロ探索 (MCS)

- 価値ベースの強化学習手法の 1 つ

- 1 エピソード終了後に以下の式で Q 値を更新

エピソード中のステップ $t = 0 \sim \mathrm{T} - 1$について

$$Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha G_t$$
$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ \ldots + \gamma^{\mathrm{T}-t-1} r_T$$

発表の流れ

- はじめに
- 要素技術
- 構築環境
- 実験
- 結果
- まとめと今後の課題

トレーディングカードゲーム (TCG)

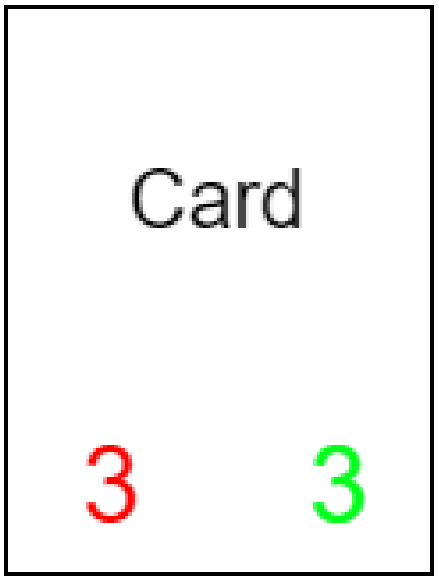
- 2 人のプレイヤーからなる
- 先攻と後攻に分かれ,
  ターン制で進行

マジック：ザ・ギャザリング.新たな旗のもとで.
2017. https://mtg-jp.com/reading/publicity/0019775/

- 各プレイヤーは異なる複数のユニットからなる
  デッキを持つ
- 相手プレイヤーの手札など一部の情報は観測できない
  (不完全情報ゲーム)

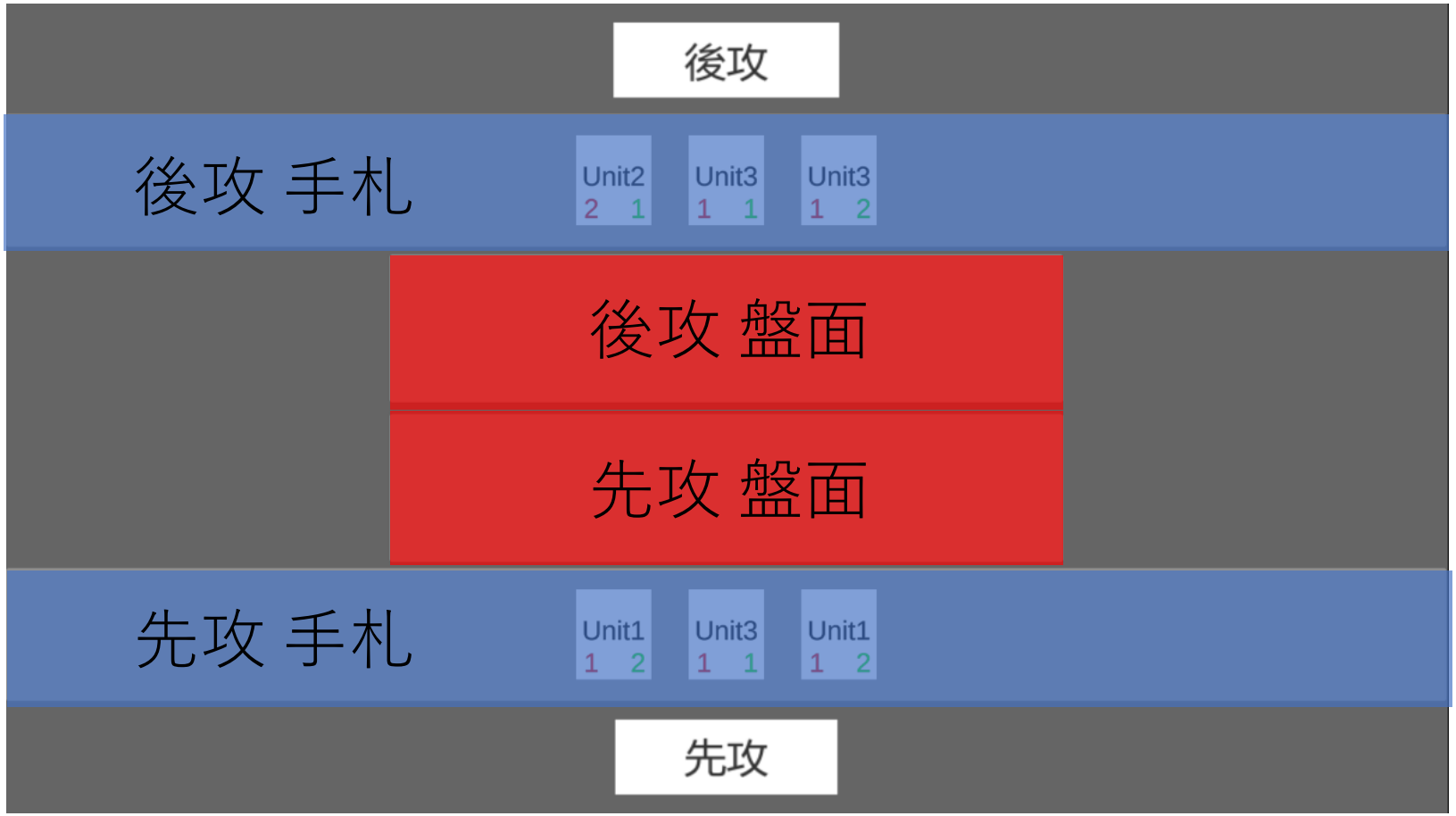
カードゲーム型対戦環境

- 2 人のプレイヤーからなる
- プレイヤーは複数のカードからなるデッキを持つ
- カードは攻撃力, HP を持つ



用語説明

- 各プレイヤーは手札, 盤面を持つ



用語説明

- ドロー

デッキからカードを取り出し, 手札に加える操作

- プレイ

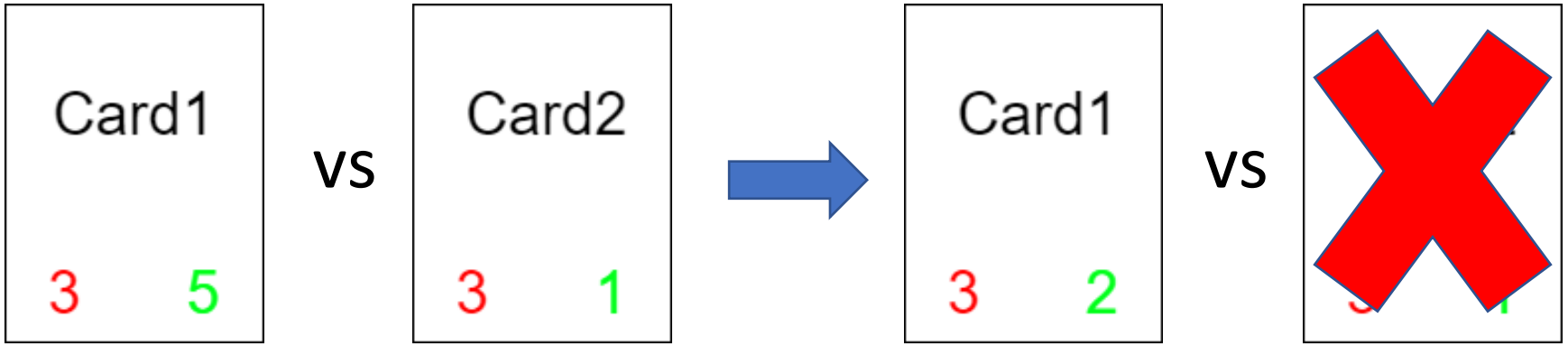
手札から盤面にカードを出す操作

- デッキ切れ

ゲーム中にデッキのカードが無くなる状態

カードの攻撃処理

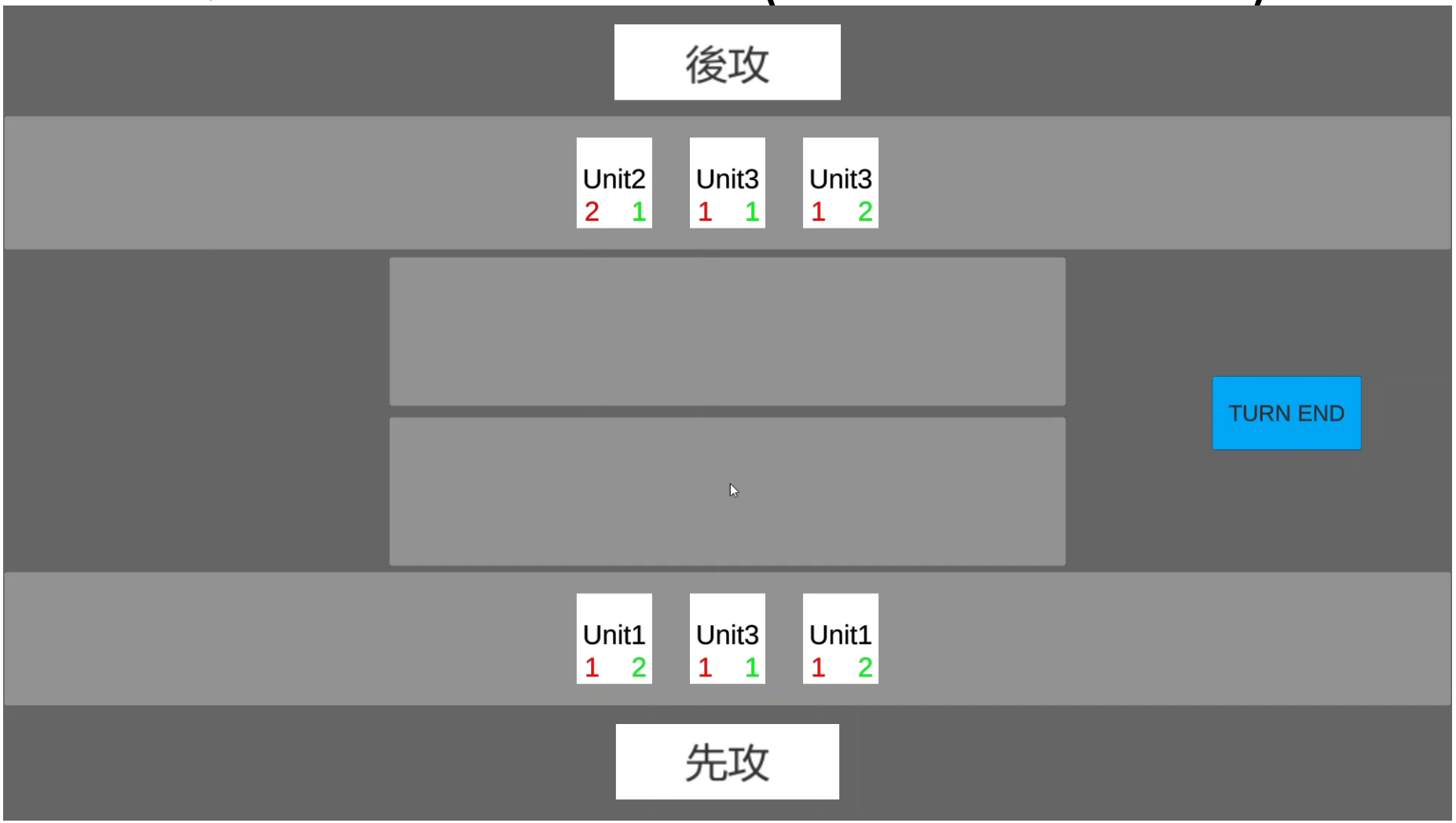
- 盤面にあるカードは相手の盤面のカードに攻撃することができる
- プレイされた次のターンから攻撃できる
- 攻撃したカードは攻撃対象から反撃を受ける



ゲームフロー

1. 各プレイヤーはデッキをシャッフル
2. 各プレイヤーは初期手札として 3 枚ドロー
3. 先攻プレイヤーの行動
4. 後攻プレイヤーはカードを 1 枚ドローして行動
5. 先攻プレイヤーはカードを 1 枚ドローして行動
6. 4 , 5 の繰り返し
7. どちらかがデッキ切れの状態でドローしようとしたら終了

ゲームフロー(プレイデモ)



※実際には相手の手札の情報は観測できない

発表の流れ

- はじめに
- 要素技術
- 構築環境
- 実験
- 結果
- まとめと今後の課題

実験

- 後攻プレイヤーを学習
- MCS と DQN で同程度学習
  ⇒ 10000 回ゲームを実行し, 勝率を計算
- 学習時の獲得報酬の推移を記録

対戦相手の行動ルーチン

1. ターンが回ってくると 1 枚カードをプレイ
2. 自盤面にあるカードすべてについて

   If 敵盤面にカードがある

   ⇒ ランダムに選んで攻撃

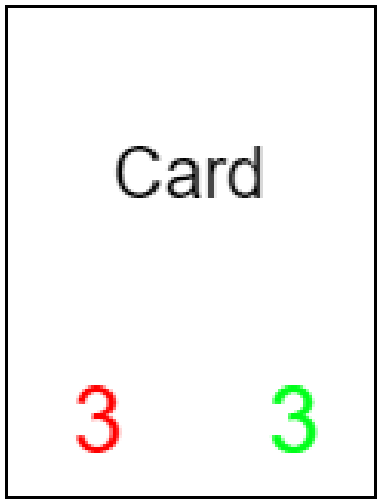
   Else

   ⇒ 何もしない

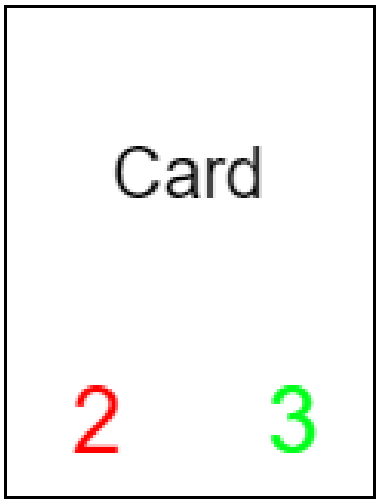
3. ターンを終了

各プレイヤーのデッキ

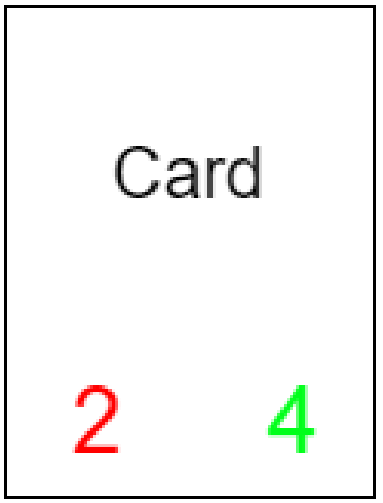
学習側も対戦相手も同じデッキを用いる



×5



×5



×5

勝利条件

ゲーム終了時に判定

(学習側盤面の枚数) > (対戦相手盤面の枚数)

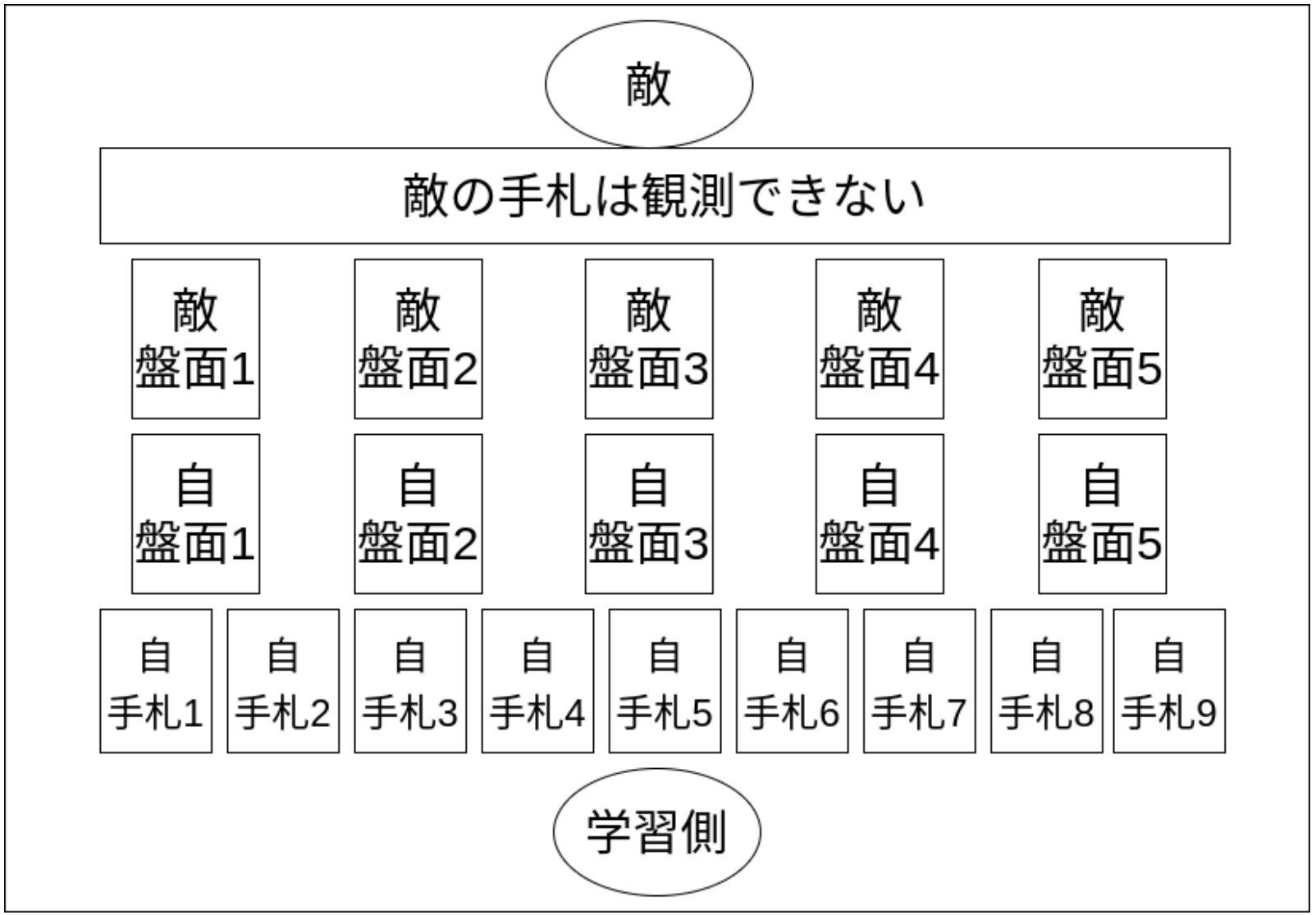
⇒ 学習側の勝利

(学習側盤面の枚数) ≤ (対戦相手盤面の枚数)

⇒ 対戦相手の勝利

行動空間, 状態空間の定義

以下の通り手札と盤面の枚数を決めておくことで定義可



状態空間

パラメータの説明	次元	値域
自手札 1 ～ 9 の HP , 攻撃力	18	0 ～ 20
自盤面 1 ～ 5 の HP , 攻撃力	10	0 ～ 20
敵盤面 1 ～ 5 の HP , 攻撃力	10	0 ～ 20
自盤面 1 ～ 5 が行動可能かどうか	5	0 ～ 1
両デッキ残り枚数	2	0 ～ 15
計	**45**	

行動空間

パラメータの説明	次元
手札 1 ～ 9 を盤面に出す	9
手札 1 ～ 9 を盤面に出さない	9
盤面 1 で敵盤面 1 ～ 5 を攻撃 or 何もしない	6
盤面 2 で敵盤面 1 ～ 5 を攻撃 or 何もしない	6
盤面 3 で敵盤面 1 ～ 5 を攻撃 or 何もしない	6
盤面 4 で敵盤面 1 ～ 5 を攻撃 or 何もしない	6
盤面 5 で敵盤面 1 ～ 5を攻撃 or 何もしない	6
計	**48**

報酬の定義

- 1 ステップ終了時

$$\mathrm{reward} = 0.0$$

- 1 エピソード終了時

$$\mathrm{reward} = \begin{cases} 1.0 & \text{(学習側勝利)} \\ -1.0 & \text{(対戦相手勝利)} \end{cases}$$

パラメータ (DQN)

パラメータ	値
割引率 γ	0.99
全結合層の活性化関数	ReLU
全結合層の次元	64
最適化アルゴリズム	Adam
方策	ε – greedy
ε	0.1
Experience Replay 開始ステップ	1.0×10^4
Target Network 更新重み	0.5
学習ステップ数	5.0×10^6

パラメータ (MCS)

パラメータ	値
学習率 α	0.5
割引率 γ	0.99
学習エピソード数	8.5×10^4

エピソード中のステップ $t = 0 \sim \mathrm{T} - 1$ について

$$Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha G_t$$
$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ \ldots + \gamma^{\mathrm{T}-t-1} r_T$$

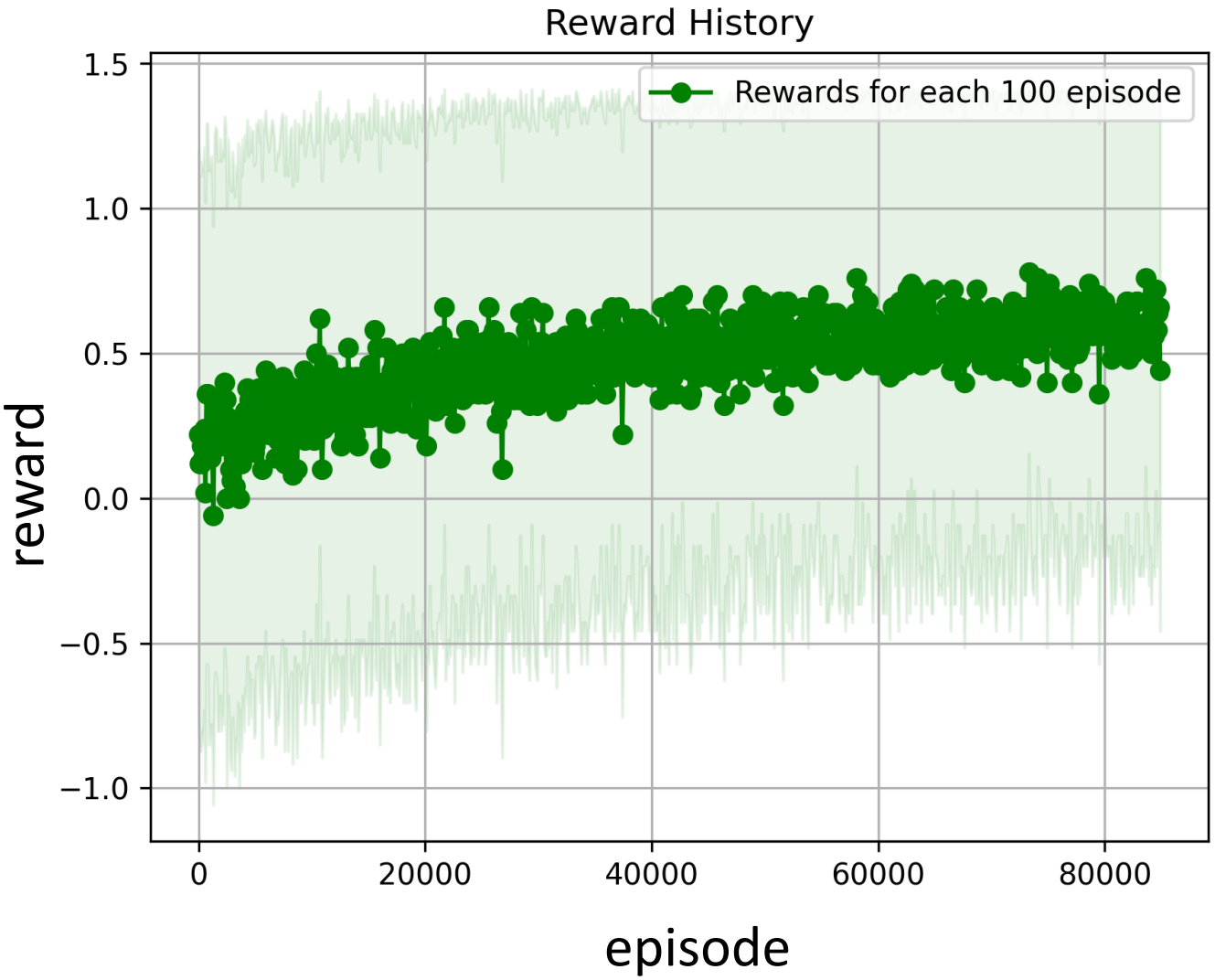
発表の流れ

- はじめに
- 要素技術
- 構築環境
- 実験
- 結果
- まとめと今後の課題

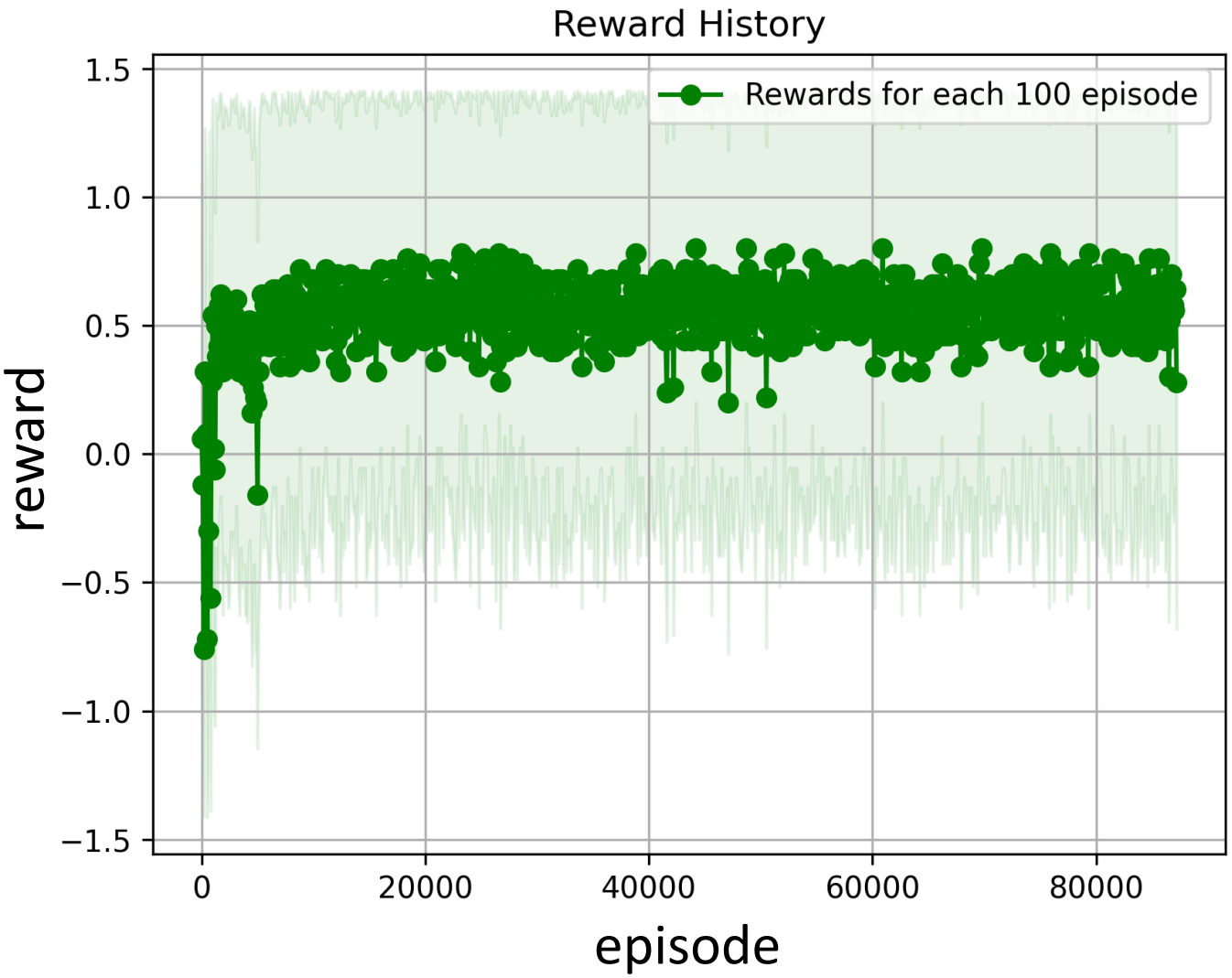
結果

手法	勝率
DQN	**0.9069**
MCS	0.7257
対戦相手と同じ戦略	0.1294

MCS の学習過程


Reward History
Rewards for each 100 episode
reward
episode
1.5
1.0
0.5
0.0
−0.5
−1.0
0
20000
40000
60000
80000

DQN の学習過程


Reward History
Rewards for each 100 episode
reward
episode
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
20000
40000
60000
80000

発表の流れ

- はじめに
- 要素技術
- 構築環境
- 実験
- 結果
- まとめと今後の課題

まとめ

- 簡易的なルールのカードゲーム型対戦環境を構築できた.

- 構築環境に対して DQN や MCS といった強化学習手法を適用できた.

本研究の目的

- カードゲーム型対戦環境の構築
- 構築環境への深層強化学習の適用
- 構築環境のゲームバランスの調整

  ⇒ 未達成

今後の課題

- ゲームバランス調整の実験

構築環境におけるゲームバランス調整に取り組む

- 環境のルール改良

ゲームデザインが詰めきれていない

⇒ TCG に存在するコストといった仕様を取り入れ,
戦略性の高いカードゲーム型対戦環境を構築する

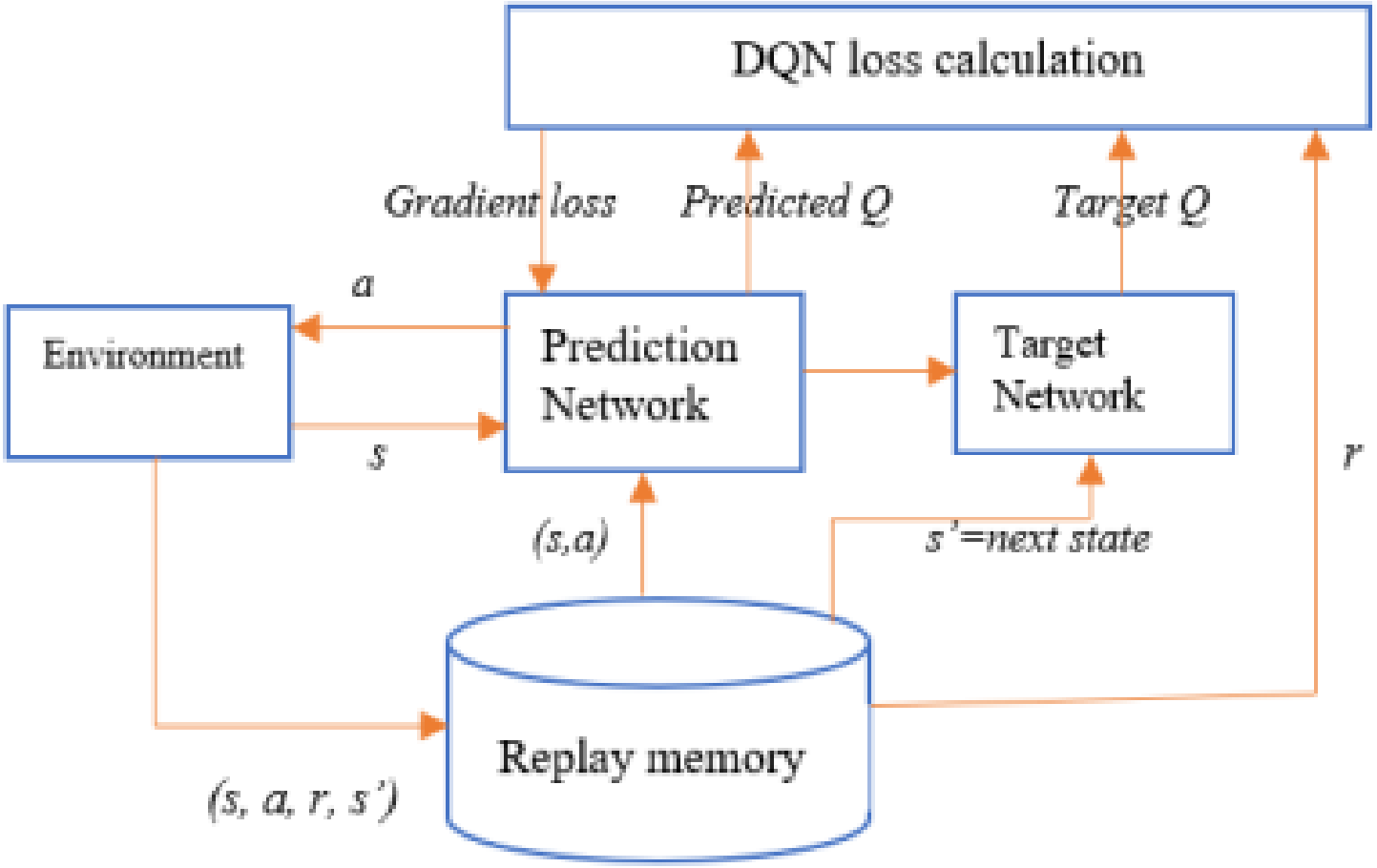
ご清聴ありがとうございました.

Experience 開始ステップについて

Exprerience Replay

- $\{s_t, a_t, r_t, s_{t+1}\}$ の組をReplay Memoryに保存
- ランダムにサンプリングして 1 つのバッチとして NN で学習することで局所解に陥ることを減らす
- 序盤に探索した解をサンプリングしてこないように 10000 に調整した
- その分局所解に陥るリスクは増えるが,その点は
- エピソードごとにデッキのシャッフルのランダム要素があるためあまり考慮しなくてもよいと考えた.

DQN



Arwa, Erick & Folly, Komla. (2020). Reinforcement Learning Techniques for Optimal Power Control in Grid-Connected Microgrids: A Comprehensive Review. IEEE Access. 8. 1-16. 10.1109/ACCESS.2020.3038735.

DQN

Algorithm 1 Deep Q-learning with Experience Replay

Initialize replay memory $\mathcal{D}$ to capacity N
Initialize action-value function Q with random weights
for episode $= 1, M$ **do**
  Initialise sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**
    With probability ϵ select a random action a_t
    otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ according to equation 3
  end for
end for

Mnih, V., Playing Atari with Deep Reinforcement Learning, arXiv e-prints, 2013.

ルールについて

- 勝利条件的に最後のターンに盤面に手札のカード全部出せばいいのでは？

⇒ その場合を防ぐために勝利条件においてカードの枚数が等しい場合に対戦相手側勝利とした