深層強化学習を用いた麻雀プレイヤの構築

清水大志[1,a)]　田中哲朗[2,b)]

概要：本研究では，麻雀で人間の知識をなるべく用いずに人間を超える実力を持つコンピュータプレイヤを作成することを目標とし，そのための第一歩として麻雀を簡略化したすずめ雀を用いて強化学習の効率を高める方法を探求する．すずめ雀は通常の麻雀から手牌や用いる牌の種類を減らし，ルールも単純化したゲームである．多人数ゲームの強化学習を行う場合，single agent の強化学習のように環境として他プレイヤを用意しなくてはいけないが，本研究では，自分の手牌のみを考慮に入れて割引累積報酬和の期待値が最も高い牌を切る一人すずめ雀プレイヤを対戦相手として強化学習を行い，一人すずめ雀プレイヤに迫る強さのプレイヤを作成できた．一方，各局の点数の最大化を目指すのではく，全局を終えたときの平均順位を最小化することを目指して，Super Phoenix で提案された Global Reward Prediction による予測値を報酬に用いる試みを行ったが，平均順位の改善は達成できていない．

Building mahjong player using deep reinforcement learning

Taishi Shimizu[1,a)]　Tetsuro Tanaka[2,b)]

Abstract: In this research, we aim to create a computer player with the ability to surpass human beings. As the first step to that end, we will explore a method to improve the efficiency of reinforcement learning by using a simplified mahjong game, Suzume-Jong. Suzume-Jong is a game that reduces the number of hand tiles and tile types from ordinary mahjong and has a simplified rule. When performing reinforcement learning of a multiplayer game, it is necessary to prepare another player as an environment like the reinforcement learning of a single agent. In this research, as opponent players, we used a Suzume-jong player that selects a move that maximizes the expected value of the sum of the discounted rewards taking only his tiles into account. As a result, we succeeded in creating a player of comparable strength to the opponent's players. Next, we tried to use the predicted value by Global Reward Prediction proposed by Super Phoenix as a reward, aiming to minimize the average ranking. However, we have not achieved the improvement of the average ranking.

1. はじめに

近年，multi-player Texas holde'm [1] や StarCraft2 [2] などいくつかの多人数不完全情報ゲームにおいて，コンピュータプレイヤの作成に強化学習を用いる手法が提案されている．その中には作成されたコンピュータプレイヤが人間の上級者の実力を超えたと主張する研究もある．例えば 2019 年 8 月に発表された Super Phoenix [3]（以下 Suphx と表記する）では，ネット麻雀の天鳳[*1]においてコンピュータプレイヤで初めて十段を達成している．Suphx は学習の始めに人間の牌譜を使った教師あり学習を行なっているが，本研究では自己対戦による強化学習で人間の牌譜を使わずに同様の実力を得ることを目標としている．強化学習を用いた自己対戦には大量の計算機を使った実験が必要であり，本研究では目標への第一歩として，麻雀を単純化したすずめ雀 [4] を用い，ベースラインプレイヤとの対戦による強化学習で強いコンピュータプレイヤを作成することを試みて，強化学習の効率を高める方法を探求する．

[1] 東京大学大学院総合文化研究科
Graduate School of Arts and Sciences, The University of Tokyo
[2] 東京大学情報基盤センター
Information Technology Center, The University of Tokyo
[a)] shimizu-taishi650@g.ecc.u-tokyo.ac.jp
[b)] ktanaka@g.ecc.u-tokyo.ac.jp

[*1] https://tenhou.net/

すずめ雀は，二人から五人で行うゲームで，「麻雀で使う牌の種類や手牌の枚数が減っている」「鳴きが存在しない」といった点で，麻雀のルールを単純化したゲームである．このゲームを対象にすることによって，学習が短い時間で進み，適用手法の効果の検証が行いやすくなるという効果が期待される．麻雀を単純化したゲームとしてはミニ麻雀 [5] もあるが，すずめ雀では役が多くあり，また実際の麻雀に近くなるようゲームバランスも考慮されているため，実験環境に適していると判断した．

本研究では，手生成の特徴量（hand-crafted features）や上級者の牌譜といったゲームに関する人間の事前知識をなるべく使わずに強いすずめ雀プレイヤを作成することを目標とする．はじめに，自分の手牌のみを考慮に入れて割引累積報酬和の期待値が最も高い牌を切ることができる一人すずめ雀プレイヤを対戦相手として強化学習を行い，一人すずめ雀プレイヤに迫る強さのプレイヤを作成できた．次に，各局の点数の最大化を目指すのではく，全局を終えたときの平均順位を最小化することを目指して，Super Phoenix で提案された Global Reward Prediction による予測値を報酬に用いる試みを行ったが，平均順位の改善は達成できていない．

以下，第 2 章では強化学習を用いた麻雀プレイヤの作成に関する主要な関連研究について，第 3 章ではすずめ雀のルールについて，第 4 章では提案手法について，第 5 章ではまとめと今後の課題について述べる．

2. 関連研究

この章では，強化学習を用いた麻雀プレイヤの作成に関する主要な関連研究を紹介する．

爆打

水上ら [6] は教師あり学習と強化学習を組み合わせて麻雀プレイヤを作成する手法を提案した．この手法では，はじめに人間の牌譜を用いた教師あり学習により，手牌のみを入力としてその時の切る牌を出力する一人麻雀プレイヤを構築する．この一人麻雀プレイヤは鳴きやリーチなどの行動はできないため，一局の間に一度だけランダムに行動するといった改善を加えたプレイヤ同士を対局させて牌譜を生成した．それらを用いて，手牌から和了時の翻数を予測するモデルを強化学習により構築している．このモデルの出力と期待最終順位を用いて点数状況を考慮する麻雀プレイヤは，水上らが以前に作成した麻雀プレイヤ [7] には負け越している．原因としては，以前の麻雀プレイヤと比べて和了率が 0.01 ほど下がっており，単純な牌効率が悪化した可能性を指摘している．

この手法では手牌を 6,661,309 次元の特徴量に変換して入力として用いているが，具体的には「役牌の刻子の数」「向聴数」など自身らの麻雀に対する知識を利用して手作業で構築しているため，この特徴量は手生成の特徴量であると言える．

Suphx

Suphx [3] は Microsoft Research Asia が 2019 年 8 月に発表した麻雀プレイヤで，初めてネット麻雀の天鳳において十段を達成した．天鳳の牌譜をもとに教師あり学習を行い，自己対戦による強化学習でモデルを改善している．

Suphx には打牌モデル，リーチモデル，チーモデル，ポンモデル，槓モデルという 5 つの CNN モデルにルールベースの和了モデルを加えた 6 つのモデルで構成される．和了モデルは「和了できる時は和了する．ただし，最終局で和了してもラスになる時はあがらない．」という単純なモデルであるが，残りの 5 つのモデルでは，教師あり学習，強化学習，チューニングの三つの学習ステップを経ている．

教師あり学習においては，各 5 つのモデルにおいて天鳳の牌譜から 4000 万〜1 億局面を使用し，手牌や得点などの現在の状況を入力としてその局面での行動を学習した．次に，得られたモデルを元に自己対戦による強化学習でモデルの改善を行った．最後に，このモデルでオンライン対戦を行う時には，run-time policy adaptation と呼ばれる実際に配られた手牌にモデルをチューニングする手法を用いた上で行動を決めている．

なお，強化学習時には

(1) policy gradient algorithm
(2) global reward prediction
(3) oracle guiding

という三つの Suphx 独自に基づいた学習を行っている．(1) は，損失関数に policy のエントロピー正則化項を加えることで policy を適切に学習できるように調整する，という手法である．(2) について，麻雀においては自分から振り込んで得点を少し減らしてでも局を進めた方が良いこともあり，局ごとの収支が強化学習の良い報酬とはならない．そこで強化学習における良い報酬を出力するために，現在の局数やそれまでの累積得点といったある局での情報を表す特徴ベクトルを入力として最終結果を予測する Global Reward Predictor Φ を作る．Suphx では Φ に 2 層の Gated Recurrent Unit（GRU）と 2 層の全結合層を足したモデルを採用している．実際に得られた報酬と予測値の二乗誤差を小さくするように天鳳の牌譜から学習し，k 局目の特徴ベクトルを x^k として $\Phi(x^k) - \Phi(x^{k-1})$ を強化学習の報酬に使うというのが global reward prediction という手法である．(3) について，麻雀には山や相手の手牌など，プレイヤからは見えない情報がたくさん存在する．そのようなゲームの全ての情報にアクセスできる oracle agent を取り入れて，強化学習の学習スピードを上げるという手法である．強化学習のはじめでは oracle agent であるが，全ての情報にアクセスできない normal agent に徐々

に近づけながら学習を進める．

3. すずめ雀のルール

この章においては，本研究で対象とした麻雀を単純化したゲームであるすずめ雀のルールを説明する．すずめ雀は通常の麻雀とは以下の点で異なっている．

- プレイヤの数は 2 人から 5 人とする．
- 各プレイヤの手牌は 5 枚．
- あがりには面子を二つ作る必要がある．
- 使用する牌は索子 9 種類と字牌 2 種類（發と中）である．同一牌の枚数は 4 枚で通常の麻雀と同じだが，数牌には 1 枚ずつ赤牌が存在する．
- 中は全て赤ドラとする．
- ドラはドラ表示牌と同じ牌とする．
- 鳴き（ポン，チー，カン）は存在しない．
- 役は表 1 のものがある．それらの役に加えて手牌で構成する二面子において，順子を構成することで 1 点，刻子を構成することで 2 点を得ることができる．役満成立時には，その役満の得点が手に入る．
- あがるのに必要な最低得点は 5 点とする．
- 複数プレイヤから同時にロンされたときは，牌を捨てたプレイヤーが全ての得点を支払う．
- フリテンは存在するが，現物のみがフリテンとなる．このため通常の麻雀ではあがり牌のうち 1 種類でも自分の捨て牌にあったらロンあがりはできないが，すずめ雀では自分が捨てていない種類の牌ならロンあがりができる．
- 山札（ドラ表示牌を除く）を使い切る（2 人プレイの時は 16 巡と 1 人，5 人プレイの時は 3 巡と 3 人）と流局となり，親が変わる．
- 親があがった時は，上記の役の得点と手牌構成の得点の合計得点にさらに 2 点加算される．ただし，この得点はあがるのに必要な最低得点には含めない．
- ツモあがりは，他のプレイヤで等分とする．ただし端数は切り上げる．
- 各プレイヤは最初に 40 点ずつ持つ．得点がマイナスになってもゲームは続行する．
- 連荘は存在せず，全プレイヤが 4 回親を行う（全プレイヤの数を p とするとき，全部で $4 \times p$ 局行う）ことでゲームが終了する．
- チョンボをした時には，自分以外のプレイヤに 2 点ずつ支払って，そのままゲームを続行する．
- すずめ雀の公式ルールでは，自分の得点がマイナスになったらそれ以上支払わなくて良い．しかし本研究では自分の得点がマイナスになっても支払うことにする．

例えば 3 人対戦で親が「3477r7」（r は赤牌を表す）という手牌の時に，「5」をツモったとする．この時の得点は，タンヤオ 1 点，赤ドラ 1 点，順子 1 点（345），刻子 2 点（77r7）で，あがるために必要な 5 点を満たしている．親のあがりの場合は 2 点が加算されるので，あがり時の合計得点は 7 点となる．ツモ時はそれを他の 2 人のプレイヤで等分して端数は切り上げるため，親は子から 4 点ずつの計 8 点を獲得する．

表 1 通常役と役満の名前，得られる得点，役の説明．

通常役		
名前	得点	説明
タンヤオ	1	2 から 8 で二面子を構成
チャンタ	2	各面子に一九字牌を含む
ドラ	1	一枚につき一点
赤ドラ	1	一枚につき一点
役満		
オールグリーン	10	2, 3, 4, 6, 8, 發で 2 面子を構成
チンヤオ	15	一九字牌で二面子を構成
スーパーレッド	20	赤牌で二面子を構成

通常の麻雀には存在する「ポン，チー，カンなどの鳴きを考慮した打牌選択」をする必要がすずめ雀には無いものの，ミニ麻雀には含まれなかった「他のプレイヤからのロンあがり，他のプレイヤへの振り込みを考慮した打牌選択」や「他のプレイヤーとの得点差，順位を考慮した手作り」という要素がすずめ雀には存在する．

4. 提案手法

4.1 一人すずめ雀プレイヤの作成

自分の手牌のみから行動を決定する一人すずめ雀プレイヤを作成し，牌譜生成やベースラインプレイヤとして使用する．この一人すずめ雀プレイヤは以下のような条件のもとで，割引累積報酬和の期待値が高い牌を切ることができるようなプレイヤである．

- ドラ表示牌と自分の手牌に存在しない牌を引く確率は，どの牌もそれぞれ等確率であるとする．例えば手牌に中が 3 枚, 發が 2 枚, 1 索が 0 枚ある時，發を引く確率は中を引く確率の倍であり，1 索を引く確率は更にその倍になる．
- ツモの回数に制限はなく，あがるまで引く．
- あがり時の得点を報酬とし，あがり時以外の報酬は 0．
- 割引率 γ は 0.9 を用いる．

このタスクはは報酬はあがった時の得点 r のみのエピソードタスク（episodic task）に対応する．スタートから n 手であがった時の累積報酬和 R は割引率を γ として，

$$R = \gamma^n r$$

となる．

エピソードタスクの時には割引率を 1 に設定することもあるが，高い役であがることと早くあがることという相反する目的を達成することは麻雀の重要な特徴であり，割引率を 1 とした時には前者のみを達成することが目的となるため，割引率は 1 未満に設定する．

このタスクでは現在の巡目や捨て牌に関する情報は最善手を求めるにあたって不要である．1 枚捨てた時の手牌とドラだけの情報からから，累積報酬和の期待値を求めることができる[*2]．手牌の全状態数は 529647 となり，その期待値を Value Iteration [8] により小数点第 5 位まで求めた．累積報酬和の期待値が最大となる手を切り，期待値が最大となる手が複数ある時はそのうちの一つを乱数で選択して切り，チョンボ（点数が足りない，フリテンでない）せずにロンあがりができる時は必ずロンあがりするプレイヤを以降では一人すずめ雀プレイヤと呼ぶ．

一人すずめ雀プレイヤをランダムで行動するプレイヤと対戦させた．手牌から牌を一枚切るという行動を 1step として，2 人プレイで 3×10^6 step（対局数にすると 52845 ゲーム 422760 局）分行ったところ，平均順位が 1.00（1 位が 52832 回）であった．5 人プレイ時でも 3×10^6 step（対局数にすると 54317 ゲーム 1086340 局）分行ったところ，平均順位が 1.139（1 位が 49292 回）であり，一人すずめ雀プレイヤはランダムプレイヤよりは有意に強くなることが確認できた．またあがり回数は 2 人プレイ時で 381602 回（ツモあがりが 184165 回，ロンあがりが 197437 回），5 人プレイ時で 301255 回（ツモあがりが 40403 回，ロンあがりが 260852 回）であった．2 人プレイ時と 5 人プレイ時のあがり時の巡目とその回数をヒストグラムにしたものを，それぞれ図 1 と図 2 に示す．あがりの平均巡目は 2 人プレイで 7.58 巡，5 人プレイでは 2.66 巡であった．

なお全員が一人すずめ雀プレイヤであったときも 3×10^6 step 分（2 人プレイ時で 70079 ゲーム 560632 局，5 人プレイ時で 68058 ゲーム 1361160 局）だけ対戦を行った．この時のあがり回数は，2 人プレイ時で 279330 回（ツモあがりが 140430 回，ロンあがりが 138900 回），5 人プレイ時で 240471 回（ツモあがりが 38499 回，ロンあがりが 201972 回）であった．あがりの平均順目は 2 人プレイ時で 5.8 巡，5 人プレイ時で 2.43 巡であった．

またランダムプレイヤとの対戦成績として，あがり時の平均獲得得点は 2 人プレイ時で 8.08 点，5 人プレイ時で 6.99 点であった．2 人プレイ時と 5 人プレイ時の流局数とあがり回数，その時の獲得得点をヒストグラムで表したものを図 3 と図 4 に示す．なお，全員が一人すずめ雀プレイヤであった時，2 人プレイ時では平均獲得得点が 7.69 点，5 人プレイ時では 6.96 点であった．

4.2 強化学習によるすずめ雀プレイヤの作成

すずめ雀を強化学習で行うための環境を作成し，強化学習を行ってすずめ雀プレイヤを作成した．報酬関数には，現在の順位を順位点に変換したものと局終了時の得点を平均したものを用いた．なお麻雀でプレイヤが最大化を目指す順位点は，最終順位や最終得点によって決まる．すずめ雀のルールには順位点の記述が無いが麻雀の要素には必要なものために導入する．今回は p 人の対戦で r 位であった時，順位点は $1 + 2 \times (1 - r)/(p - 1)$ とした．強化学習アルゴリズムとしては Suphx でも使われている PPO [10] を

[*2] ロンあがりが存在しないため，自分の捨て牌に関する情報を覚えておく必要はない

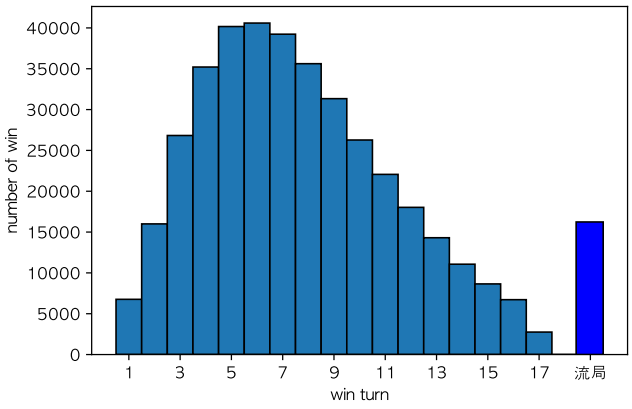

図 1　2 人プレイ時の流局数，あがり回数とその巡目．

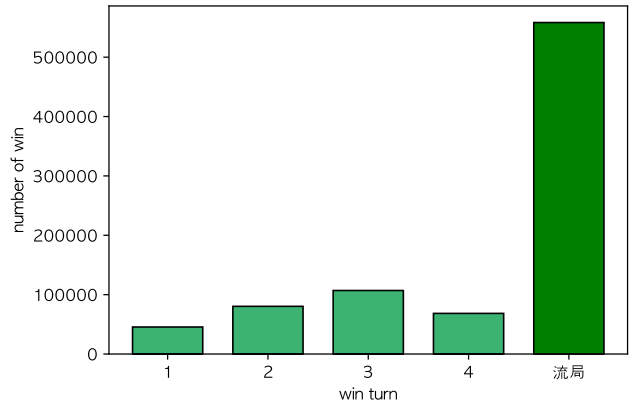

図 2　5 人プレイ時の流局数，あがり回数とその巡目．

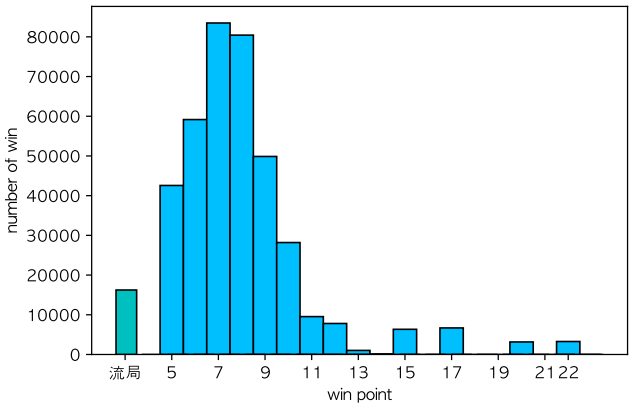

図 3　2 人プレイ時の流局数，あがり回数とその獲得得点．

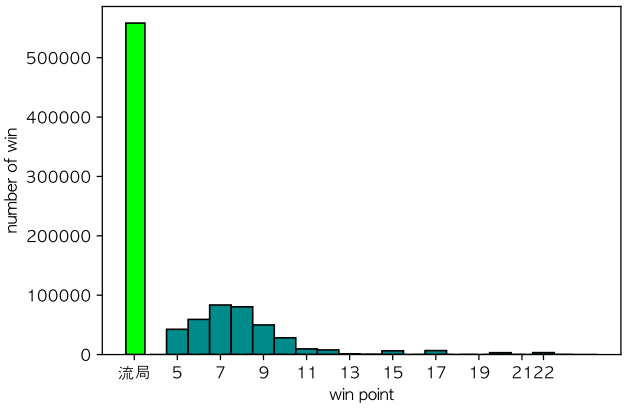

図 4　5 人プレイ時の流局数，あがり回数とその獲得得点．

使用した．モデルは OpenAI baselines [*3]のデフォルトである「Multi Layer Perceptron（MLP）」（隠れ層が 2 層で，ユニット数はそれぞれ 64）と，モデルにより表現力がある ResNet [9] を改良した「resnet」の二つのモデルを使用した．

この resnet の具体的な層の構成を図 5 に示す．Conv1d の最初の引数はカーネルサイズを，2 つ目の引数はフィルターの数を表す．その他のハイパーパラメータは tensorflow[*4]のデフォルト値に従った．ゲームを対象にした先行研究の多くでは ResNet を用いる際は res_block の数は 10 個以上とすることが多く，例えば AlphaGo Zero [11] では 19 個あるいは 39 個が用いられている．しかし，本研究では一人すずめ雀プレイヤのポリシーを教師ありで層の数の異なる resnet に学習させる予備実験を行い，5 個以上では大差がなかったので，学習時間の短い 5 個を用いた．

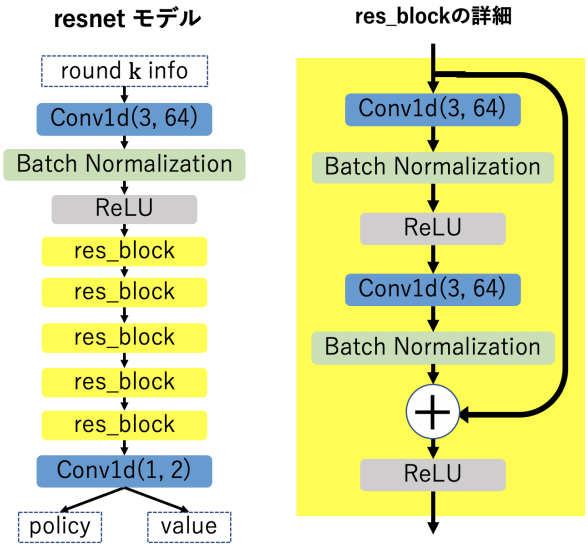

図 5　resnet モデルの各層の構成．

*3 https://github.com/openai/baselines
*4 https://www.tensorflow.org/

入力には親，現在の局数，ドラ，自分の手牌，各プレイヤの累積得点，各プレイヤの捨て牌を使用しており，次のように表現する．

自分の手牌とドラに関しては，13 行 8 列，要素は 0 か 1 の二値の行列を用いる．行は牌の種類を表すが，9 索と發，發と中の間には 1 行入れるため 13 行となっている．その行にあたる牌が手牌に i 枚存在した時，1 列目から i 列目までを 1 とし，それ以外を 0 とする．ただし $i = 0$ の時は，1 から 4 列目まで全て 0 とする．5 列目は，その行にあたる牌で，かつ赤ドラとなっている牌を持っている時には 1，それ以外では 0 とする．6 列目に関しては，その行にあたる牌がドラ表示牌であったら 1，それ以外では 0 とする．7 行目に関しては，その行にあたる牌の赤牌がドラ表示牌であったら 1，それ以外では 0 とする．8 行目に関しては，10 行目と 12 行目は 0 とし，それ以外は 1 とする．ドラが 2 索で，自分の手牌が「23r344 發」であった時の入力の具体例を図 6 にあげる．

		1枚以上	2枚以上	3枚以上	4枚	赤を含むか	ドラか	赤牌がドラか	牌かどうか
牌の種類	1	0	0	0	0	0	0	0	**1**
	2	**1**	0	0	0	0	**1**	0	**1**
	3	**1**	**1**	0	0	**1**	0	0	**1**
	4	**1**	**1**	0	0	0	0	0	**1**
	5	0	0	0	0	0	0	0	**1**
	⋮								
	9	0	0	0	0	0	0	0	**1**
	blank	0	0	0	0	0	0	0	0
	發	**1**	0	0	0	0	0	0	**1**
	blank	0	0	0	0	0	0	0	0
	中	0	0	0	0	0	0	0	**1**

図 6　ドラが 2 で自分の手牌が「23r344 發」であった時の入力例．

現在の局数，親に関しては one-hot エンコーディングとする．ゲームに参加する人数を p として，現在の局数は $4 \times p$ 次元，親は p 次元とし，当てはまる要素に 1 を，それ以外の要素には 0 をいれる．各プレイヤの累積得点に関して，0 から 100 点については 5 点ずつのバケットに分割して当てはまる要素を 1，それ以外を 0 とする．0 点以下と 100 点以上はそれぞれ一つの次元でまとめるため，プレイヤ 1 人あたり合計で 22 次元となる．各プレイヤの捨て牌に関しては，その行にあたる牌が捨ててあれば 1，それ以外は 0 とする．ただし，親，現在の局数，各プレイヤの累積得点に関しては 13 行に渡って同じものを並べることで，全部で $13 \times (27 \times p + 8)$ の行列で入力を表せるようになる．

プレイヤはあがれる時には必ずあがるものとし，自分以外のプレイヤは一人すずめ雀プレイヤと同じ行動を取るとした．プレイヤが行動をしてから，もう一度自分の手番が回ってくるまでを 1step とし，3×10^7 step 学習させた．学習時の resnet モデルでの policy loss と value loss について，2 人プレイ時を図 7 に，5 人プレイを図 8 に示す．

得られたモデルにおいて，強化学習で得られたプレイヤを 3×10^6step 対戦させた時の平均順位を表 2 に示す．こ

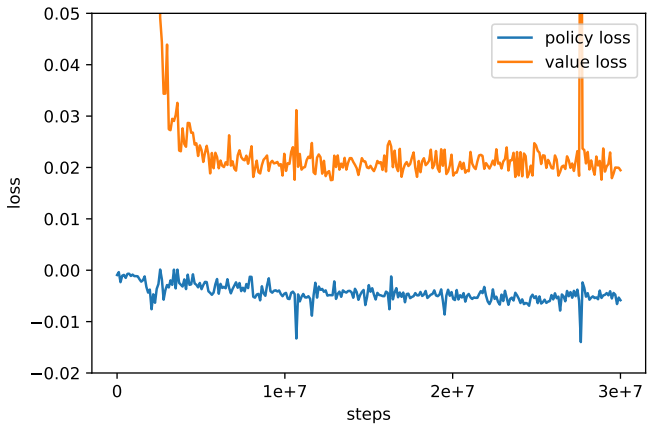

図 7　2 人プレイ時の resnet モデルでの policy loss と value loss.

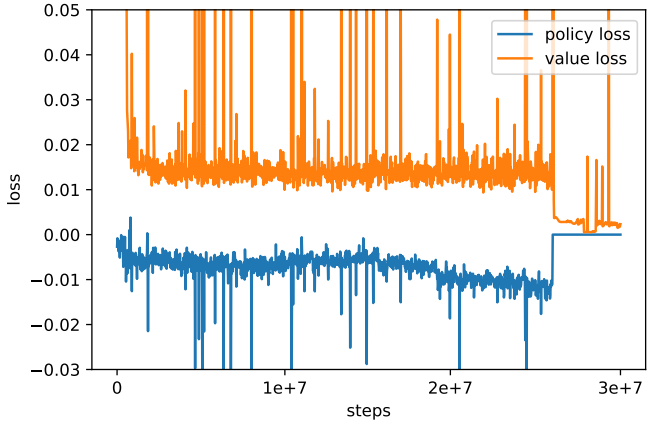

図 8　5 人プレイ時の resnet モデルでの policy loss と value loss.

の step 数は約 65000 ゲームに相当する．

表 2　MLP モデル，resnet モデルにおける平均順位．

設定＼モデル	MLP	resnet
二人プレイ	1.748	**1.528**
五人プレイ	3.501	**3.034**

2 人プレイでは平均順位が 1.5 で同等，5 人プレイでは平均順位が 3 で同等であると言える．この実験結果において，MLP を使用したモデルはどちらの実験設定においても一人すずめ雀プレイヤに対して負け越しているが，resnet を使用したモデルは一人すずめ雀プレイヤと同程度の実力を得たことがわかる．MLP はモデルの表現力が十分でないが，resnet はある程度の表現力を持つからであると考えられる．

4.3　Global Reward Predictor の作成

次に得られたモデルを改良することを目標として，強化学習における良い報酬を出力するために最終結果を予測する Global Reward Predictor Φ を作る．この予測器の入力には，現在の局数 k，親，それまでの累積得点，その局の収支を含む特徴ベクトル x^k を用いる．なお Suphx ではそれらに加えて供託や連荘の回数という情報も含むが，すずめ雀においてはそれらは存在しないので含んでいない．具体的な表現方法としては以下である．

現在の局数や親というカテゴリカルな情報は，強化学習時の入力と同様に表す．その局までの累積得点やその局の得点はバケットに分割して当てはまる要素を 1，それ以外を 0 とする．累積得点については強化学習時と同様に表す．その局の得点については −21 点以下から 19 点以上までを 2 点刻みのバケットに分割するため，この情報は合計で 22 次元となる．

使用したモデルは図 9 のように，Suphx で使われたモデルだけでなく，RNN model と Dense model を追加した．GRU と RNN の引数はユニット数を，Dense の引数はユニット数と活性化関数を表す．RNN model に関しては GRU よりも単純な構造で置き換えた時にどうなるのかを調べるため，また Dense model に関しては，特徴ベクトル x^k には以前の局を踏まえた情報が含まれており，reccurent なモデルを使用しなくても良い可能性があると判断したためである．前述の一人すずめ雀プレイヤ同士の対戦のログ 3×10^6step 分（2 人プレイでは 70079 ゲーム分，5 人プレイでは 68058 ゲーム分）を学習に使用した．この学習では，損失関数は Suphx と同じ二乗誤差を使用する．

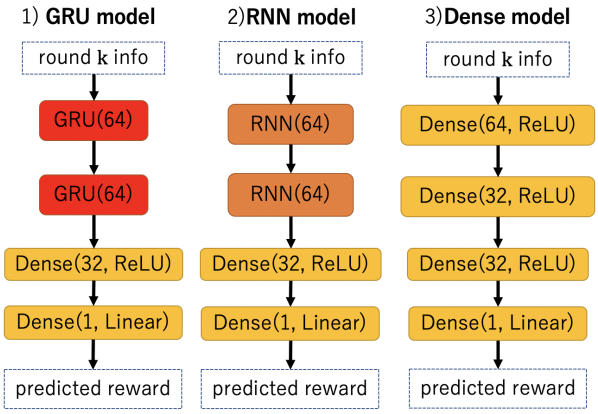

図 9　Global Reward Predictor で使用したモデルの概要.

全データのうち 90 %を訓練データ，残りをバリデーションデータとした．学習時のハイパーパラメータのうち，エポック数は 100，バッチサイズは 128 とした．残りのハイパーパラメータは keras [*5]のデフォルト値に従った．

学習時の結果を図 10 と図 11 示す．図 10 では教師あり学習時の loss を，図 11 には validation loss を示す．

loss はどの手法においても下がっているが，一番下がったのは先行研究と同じモデルである GRU model であった．

[*5] https://keras.io/ja/

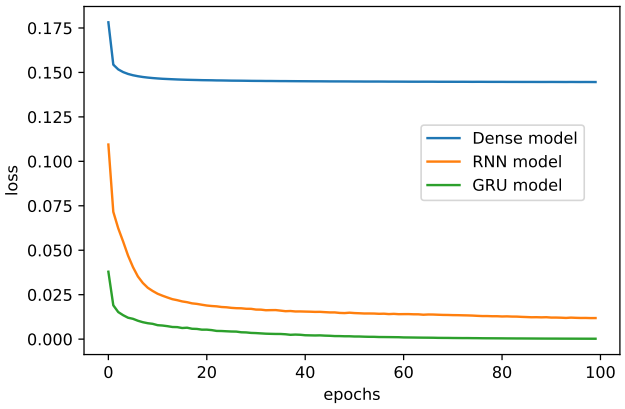

図 10　Global Reward Predictor の学習時の loss.

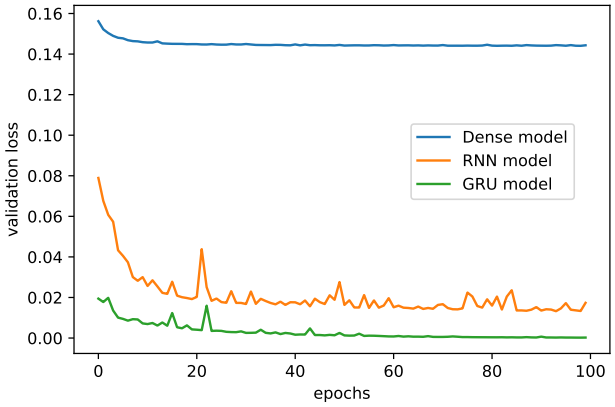

図 11　Global Reward Predictor の学習時の val_loss.

k 局目の情報である特徴ベクトル x^k には，それまでの累積得点，現在の親，現在の局数という情報が含まれているので，1 から $k-1$ 局までの情報を含んでいる．しかし，時系列データであることは変わりなく，RNN model や GRU model といった reccurent な情報を扱えるモデルの方が，loss が下がりやすいという結果であった．

そこで，Global Reward Predictor の各モデルを報酬に使用した強化学習によって，resnet のすずめ雀プレイヤを構築した．報酬以外の条件は先ほどと同じにする．得られたモデルを 3×10^6step だけ一人すずめ雀プレイヤと対戦させた．2 人プレイ時の平均順位を表 3 に示す．なお表には比較のために MLP モデルを使用したすずめ雀プレイヤの平均順位も比較のために表示している．

表 3　2 人プレイ時の各実験設定におけるすずめ雀プレイヤの平均順位.

設定＼報酬	得点と順位点	Dense model	RNN model	GRU model
MLP	1.748	**1.708**	1.718	1.715
resnet	**1.528**	1.581	1.552	1.530

Global Reward Predictor が MLP という単純なモデルにおいてはすずめ雀でも効果的であったということ，また Suphx で提案されているモデル以外の簡単なモデルでも有効であることが分かった．しかし，ある程度の表現力をもつ resnet を使用した場合は，平均順位の改善には至らなかった．

この理由は複数考えられる．一つ目は Global Reward Predictor を使用しない場合，報酬に現在の順位を順位点に変換したものを使用しているが，これが強化学習においてある程度良い報酬になっていたという可能性である．この強化学習の目標は順位点の獲得であり，ある局における現在の順位は最終順位と大きく関係があるはずである．二つ目として，一人すずめ雀プレイヤが十分に強いという可能性である．一人すずめ雀プレイヤ同士を対戦させた時，自分が一度行動するのを 1 巡として，二人プレイ時では平均 5.8 巡，五人プレイ時では平均 2.43 巡で 1 局が終了する．すずめ雀は通常の麻雀に比べて 1 局の間にできる行動が少ない．一人すずめ雀プレイヤは自分の手牌しか考慮に入れず，最善のあがりを目指すプレイヤであるが，それがすずめ雀にとっては最適な戦略であるという可能性が考えられる．

5.　まとめと今後の課題

本研究では，手生成の特徴量や上級者の牌譜といったゲームに関する人間の事前知識を使わずに深層強化学習を用いたすずめ雀プレイヤを構築した．強化学習に用いるモデルの表現力が十分でなく，まだ改善の余地があるようなすずめ雀プレイヤにおいては，Global Reward Predictor が効果的であることがわかったが，強化学習のモデルに resnet を使用し，すでにある程度の強さをもつプレイヤにとっては Global Reward Predictor が効果的な手法とは言えない可能性があることがわかった．

麻雀では，様々な要因によって学習が困難になる．例えば「プレイヤから見えない牌が多く，報酬関数を作るのが難しい」や「鳴きで順番が変わるのでゲーム木のサイズが膨大となり，MCTS などの従来手法が使えない」などである．

上記の問題を解決する手法として，Suphx で提案されている他の学習手法やそれを改善した手法があり，今後はそれをすずめ雀で評価していきたい．すずめ雀で十分な効果が得られた手法は，実際の麻雀にも適用できる可能性がある．また本研究の内容は乱数の初期シードの影響をうける．今回は一つのシードのみでしか実験を行っておらず，複数シードで実験をするということも今後の課題としたい．

なお，本研究で用いたプログラムは GitHub(`https://github.com/minnsou/suzume-jong`) で公開している．

謝辞　本研究は JSPS 科研費 18K11600 の助成を受けて行われた．

参考文献

[1] N. Brown and S. Tuomas, "Superhuman AI for heads-up no-limit poker: Libratus beats top professionals," *Science*, vol. 359, no. 6374, pp. 418-424, 2018.

[2] O. Vinyals, *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350-354, 2019.

[3] J. Li, *et al.*, "Suphx: Mastering Mahjong with Deep Reinforcement Learning," March 2020. Accessed on: July 25, 2020. [Online]. Available: `https://arxiv.org/abs/2003.13590`

[4] K.Maruta and T.Shimozaki, "Suzume-Jong," Accessed on: May 15, 2020. [Online]. Available: `https://sugorokuya.jp/p/suzume-jong/`

[5] 清水大志 and 田中哲朗, "麻雀のポリシー関数に適したネットワークモデルの構築と評価," ゲームプログラミングワークショップ 2019 論文集, pp. 165-171. 2019.

[6] 水上直紀 and 鶴岡慶雅, " 強化学習を用いた効率的な和了を行う麻雀プレイヤ," ゲームプログラミングワークショップ 2016 論文集, pp. 81-88. 2016.

[7] 水上直紀 and 鶴岡慶雅, " 期待最終順位に基づくコンピュータ麻雀プレイヤの構築," ゲームプログラミングワークショップ 2015 論文集, pp. 179-186. 2015.

[8] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," MIT press, 2018.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778, 2016.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," July 2017. Accessed on: May 17, 2020. [Online]. Available: `https://arxiv.org/abs/1707.06347v2`

[11] D. Silver, *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354-359, 2017.

