Expectimax探索結果から方策と価値を学習する（ミニ）2048向けAlphaZero型強化学習

金子　智弥[1,a)]　松崎　公紀[1,b)]

概要：2048 のプレイヤでは，価値関数を学習して探索に用いる方法が多く研究されている．本研究では，探索の結果をもとに学習すること，および，学習した方策を行動選択に利用することが性能改善につながるかを調べる．具体的には，AlphaZero の方策・価値学習の枠組みに着想を得て，深さ 2 の Expectimax 探索の結果から方策と価値を学習する方法を構成した．3×3 盤面のミニ 2048 を対象とし，学習目標の選定，学習経験の生成，学習結果の利用という三つの観点から評価した．価値教師・方策教師・学習時の行動源を組み合わせた 40 条件を各 5 シードで学習し，各モデル・評価方式を 10,000 ゲームで評価した．その結果，方策と価値の両教師を方策重み付き平均で計算し学習時の着手を最大値方式の探索で選ぶ条件では，方策単独によるプレイで 5189.6 点，方策と深さ 3 の行動価値の積によるプレイで 5222.4 点を得た．価値のみを学習するベースラインの深さ 3 によるプレイでの 5080.5 点を上回り，方策利用による改善を確認した．さらに，代表的なプレイヤの実際の着手を完全解析値で分析した．その結果，最高得点を達成した上記プレイヤは，望ましい評価値の分布を持っていた．

Learning Policy and Value from Expectimax Search for AlphaZero-Style Reinforcement Learning in (Mini-)2048

Tomoya Kaneko[1,a)]　Kiminori Matsuzaki[1,b)]

Abstract: We investigate whether learning of evaluation functions from tree search and the use of learned policies improve computer players for 2048. Inspired by AlphaZero, we train policy and value networks from depth-two Expectimax search on Mini2048. We examine three design choices: learning targets, generation of training experience, and use of the trained networks. Forty configurations are trained with five seeds each, and each model is evaluated over 10,000 games with each applicable playing method. A configuration using policy-weighted targets for both networks and maximum-backup search for training actions achieves mean scores of 5189.6 with policy-only play and 5222.4 using the product of policy probabilities and depth-three action values. Both exceed the mean score of 5080.5 achieved by the value-only training baseline with depth-three search. We also analyze actual decisions against exact optimal values. The result shows that the best evaluation functions above have desirable distributions of evaluation values.

1. はじめに

2048 は，プレイヤによる決定的な操作と新しいタイルのランダムな出現が交互に起こる確率的一人ゲームである．強いプレイヤを作る方法として，盤面の価値関数を強化学習し，その値に基づいて手を選ぶ方法が研究されてきた．特に，N タプルネットワークによる価値関数と Expectimax 探索を組み合わせる方法が発展している [1], [2]．ニューラルネットワークを用いた価値関数でも，探索との組み合わせによる性能改善が報告されている [3].

これまでの 2048 の研究では，価値関数のみを用いるプレイヤが中心であり，方策を学習して行動選択に用いる例は限られていた．本研究では方策と価値関数を学習する新

[1] 高知工科大学大学院工学研究科
Graduate School of Engineering, Kochi University of Technology
[a)] 295133i@gs.kochi-tech.ac.jp
[b)] matsuzaki.kiminori@kochi-tech.ac.jp

© 2026 Information Processing Society of Japan

しい手法を提案し，そうして得られた方策の利用による性能改善を検証する．

方策と価値を学習する手法として，AlphaZero が広く知られている．AlphaZero は，モンテカルロ木探索で得た情報から方策と価値関数を学習する [4]．実際に 2048 に対して AlphaZero 系の手法が成功することも報告されている [5], [6]．一方，2048 では，学習した価値関数と Expectimax 探索を組み合わせた高性能なプレイヤが報告されている [2]．そこで本研究では，AlphaZero の方策・価値学習の枠組みを Expectimax 探索と組み合わせ，その探索結果から方策と価値を学習する AlphaZero 型の手法を構成する．

具体的には，強化学習による方策と価値関数の学習，およびプレイヤの設計について，以下の三つの観点から検討する．

(1) **学習目標の選定：** Expectimax 探索木のどの値を方策と価値関数の教師として用いるか．
(2) **学習経験の生成：** Expectimax 探索木のどの値を学習時の行動選択に用いるか．
(3) **学習結果の利用：** 学習によって得られた方策と価値関数を，学習後のプレイヤでどのように利用するか．

現実的な時間で学習が行えるよう，学習時における Expectimax の探索の深さは 2 に固定する．

実験には，完全解析値を参照できる 3×3 盤面のミニ 2048[7] を用いる．学習目標の選定法と学習経験の生成法を変えた 40 条件を各 5 シードで学習し，それぞれについて学習結果の利用法を変えてゲーム得点を比較する．さらに，各プレイヤが訪れた盤面での着手と探索評価値を完全解析値に照らして調べ，実プレイで生じた損失の特徴や評価値そのものの分布を分析する．

2. 関連研究と本研究の位置づけ

2.1 2048 における価値学習と探索の発展

2048 の強化学習プレイヤでは，N タプルネットワークによる価値学習と Expectimax 探索を組み合わせる方法が発展してきた．Szubert と Jaśkowski[1] は，操作直後・新タイル生成前の盤面である afterstate の価値を Temporal Difference（TD）学習し，即時報酬と推定価値に基づいて手を選ぶ方法を示した．

その後，Wu ら [8] と Yeh ら [9] は，進行段階ごとに価値関数を学習する多段階 TD 学習により，終盤の学習不足を改善した．Matsuzaki[10] はタプル集合の系統的な選択を検討し，Jaśkowski[11] は重みごとの学習率を適応させる Temporal Coherence（TC）学習などを導入した．

Guei ら [2] は，重みを高い値で初期化して未訪問・低頻度の局面への探索を促す楽観的初期化と，TD 学習後に TC 学習でネットワーク重みを調整する方法を提案した．これに多段階学習，Expectimax 探索，tile-downgrading といった手法を組み合わせたプレイヤは，当時の最高性能を達成した．

2.2 探索結果を用いた学習とその課題

2048 においても，AlphaZero および Stochastic MuZero により，探索と方策・価値学習を組み合わせる方法の有効性が示されている [5]．しかし，それらよりも計算量を抑えた探索を学習へ取り入れる際には，学習目標や経験生成の設計によって，評価値の過大評価や性能低下が生じる場合がある．

渡邊 [12] は，Monte-Carlo Softmax 探索の内部ノードの評価値を用いた学習を試みたが，評価値が安定せず，既存の TD 学習を上回る結果は得られなかった．また，Wang[13] は，2-ply 探索で着手を選ぶ学習条件において，著しい過大評価と低いプレイ性能を報告している．これに対し本研究は，学習目標の構成と経験生成時の行動選択を分けて設計し，網羅的に評価することに新規性がある．適切な組み合わせの下で，Expectimax 探索の結果を用いた学習が，従来の TD 学習を改善しうることを示す．

3. (ミニ)2048 とその探索木

2048 では，上下左右のいずれかの方向へタイルを移動させ，同じ値のタイルを合成する．報酬は，その操作で合成して得られたタイルの値の合計である．操作の後，空きマスの一つが一様に選ばれ，値 2 のタイルが確率 0.9，値 4 のタイルが確率 0.1 で生成される．合法手がなくなるとゲームが終了する．ミニ 2048 では，このゲームを 3×3 盤面で行う．

時刻 t に行動を選ぶ状態を S_t，合法手集合を $\mathcal{A}_t$ とする．行動 a_t による移動・合成の直後で，新タイル生成前の盤面を afterstate S'_t，その操作の報酬を r_t とする．1 手の遷移は

$$S_t \xrightarrow{a_t, r_t} S'_t \xrightarrow{\text{chance}} S_{t+1} \tag{1}$$

である．S_t から S'_t への遷移と報酬は行動が決まれば一意であり，S'_t から S_{t+1} への遷移は確率的である．以降，候補手 a に対応する afterstate を $S'_t(a)$ と書く．

本研究ではこれらの状態遷移に基づき，方策関数を状態 S_t に対して，価値関数を afterstate S'_t に対してそれぞれ定義する．方策関数 $p(S_t, a) = \pi_\phi(a \mid S_t)$ は S_t を入力し，合法手上で正規化した選択確率を出力する．価値関数 $V_\theta(S'_t)$ は S'_t を入力し，その後に得られる得点を予測する．S'_t へ到達する際に既に得た報酬 r_t は，この継続価値に含めない．

Expectimax の探索木では，プレイヤ節点で最大値を取り，確率節点で生成確率による平均を取る．深さ d の探索木は，最大 d 回の操作と $d-1$ 回の新タイル生成を展開して得られる木構造とし，その葉は afterstate となる．

深さ 1 の行動価値を，即時報酬と操作直後の afterstate の価値から

© 2026 Information Processing Society of Japan

$$v_1(S_t, a) = r_t(a) + V_\theta(S'_t(a)) \tag{2}$$

と定める．この方法では，操作直後の afterstate を価値関数に入力し，その後に得られる得点を予測する．

深さ 2 では，各 $S'_t(a)$ から新タイル生成後の状態 S_{t+1} を列挙し，その各状態で次の合法手を展開する．葉は 2 回目の操作後の afterstate S'_{t+1} であり，その価値を V_θ で評価する．空きマス数を m とすると，新タイルの候補は最大 $2m$ 通りで，それぞれの確率は $0.9/m$ または $0.1/m$ である．afterstate S'_t の値は，これらの確率で重み付けした子節点の値の平均である．

4. Expectimax を組み合わせた学習の設計

本節では，探索結果を用いた学習における，学習目標，学習経験の生成，および学習後の行動選択を設計する．それらに用いる探索値の計算として，通常の Expectimax で用いる最大値方式に加え，方策確率による重み付き平均方式を定義する．

afterstate S' から新タイル生成によって生じる状態を S，その合法手集合を $\mathcal{A}(S)$ とする．第 3 節で定義した深さ 1 の行動価値 $v_1(S, a)$ を用いて，S' 以降の得点の推定値を次のように 2 通りの方法で定める．

$$B_{\max}(S') = \mathbb{E}_{S|S'}\left[\max_{a\in\mathcal{A}(S)} v_1(S, a)\right], \tag{3}$$

$$B_\pi(S') = \mathbb{E}_{S|S'}\left[\sum_{a\in\mathcal{A}(S)} p(S, a)v_1(S, a)\right]. \tag{4}$$

期待値はゲーム規則で定まる新タイルの生成確率について取り，終端状態では括弧内の値を 0 とする．$B_{\max}$ では最も高い行動価値を，B_π では方策確率で重み付けした平均行動価値を用いる．いずれも，S' に至る操作の報酬は含まない．

$b \in \{\max, \pi\}$ として，深さ 2 の候補手の行動価値を

$$v_2^b(S_t, a) = r_t(a) + B_b(S'_t(a)) \tag{5}$$

と表す．以後，v_1，$v_2^{\max}$，v_2^π による計算方式を，それぞれ 1-Ply，2-Max，2-PAvg と呼ぶ．PAvg は方策確率による重み付き平均（policy-weighted average）を表す．これらの名称を，学習目標と経験生成時の行動源に用いる．

4.1 方策と価値の目標値

方策教師には，2-Max または 2-PAvg で行動価値が最大となる合法手を用い，交差エントロピー損失を最小化する．

afterstate S'_t の価値教師には，新タイル生成後の次状態 S_{t+1} で実行した手の行動価値を，1-Ply，2-Max，2-PAvg のいずれかで計算して用いる．教師には勾配を流さず，価値予測との二乗誤差を最小化する．終端の教師は 0，割引率は 1（割引なし）とする．

4.2 学習エピソードの生成

教師の計算方式とは独立に，行動源を選ぶ．得られた行動価値が最大の手を選ぶ 1-Ply，2-Max，2-PAvg と，方策確率との積が最大の手を選ぶ 1-Ply×π，2-Max×π，2-PAvg×π を比較する．

4.3 学習後の行動選択

学習後は，以下の 7 方式を比較する．Policy は，方策確率 $p(S, a)$ が最大の手を選ぶ．Value(d)（$d = 1, 2, 3$）は，深さ d の Expectimax 探索による行動価値 $v_d(S, a)$ が最大の手を選ぶ．Product(d)（$d = 1, 2, 3$）は，方策確率 $p(S, a)$ と深さ d の行動価値 $v_d(S, a)$ の積が最大の手を選ぶ．

5. 評価実験

5.1 ネットワークと学習設定

方策と価値には別々の畳込みニューラルネットワーク（CNN）を用いた．入力はタイルの指数を空きマスを含む 11 種類で one-hot 化した $11 \times 3 \times 3$ テンソルである．両ネットワークとも畳込み 5 層と全結合 2 層を持つ．畳込みの出力チャネル数は 64，128，232，256，256，最初のカーネルは 1×1，以降は 2×2 とし，第 2 畳込み層に padding 1 を用いた．中間層の活性化関数は ReLU，全結合の隠れ層は 256 次元である．価値ネットワークは afterstate から 1 個の値を出力し，方策ネットワークは player state から 4 方向の logit を出力する．方策確率は非合法手を除いた softmax で求める．

各条件でモデルを 5 つの学習シードにより独立に初期化し，それぞれ学習した．最適化アルゴリズムには Adam を用い，価値・方策とも学習率 10^{-4}，ミニバッチサイズ 128 とした．256 個のゲーム環境を並列に実行し，250,000 学習ステップで学習を終了した．各環境での着手または終局後の初期化を 1 学習ステップとした．盤面と行動方向について，回転・鏡映による 8 対称をすべて学習データに用いた．評価には最終モデルを用いた．

全実験を同一の計算機・実行環境（CPU：Intel Core i7-9800X，GPU：NVIDIA GeForce RTX 2080 Ti）で行った．

5.2 比較する 40 条件と学習・評価シード数

比較する学習条件は表 1 の各組み合わせとし，「方策なし」と方策を必要とする学習の組み合わせを除く，全 40 条件を学習した．内訳は，方策を持つ 36 条件と価値のみを学習する 4 条件である．

価値教師と方策教師を変えて学習目標の影響を，行動源を変えて学習経験の生成方法の影響を調べる．さらに，第 4.3 節の行動選択方式と探索深さを変え，学習した方策と価値の利用方法を比較する．ベースラインは，価値教師 1-Ply，方策なし，行動源 1-Ply とした TD 学習である．

方策を持つ 36 条件では 7 方式，方策なしの 4 条件では

© 2026 Information Processing Society of Japan

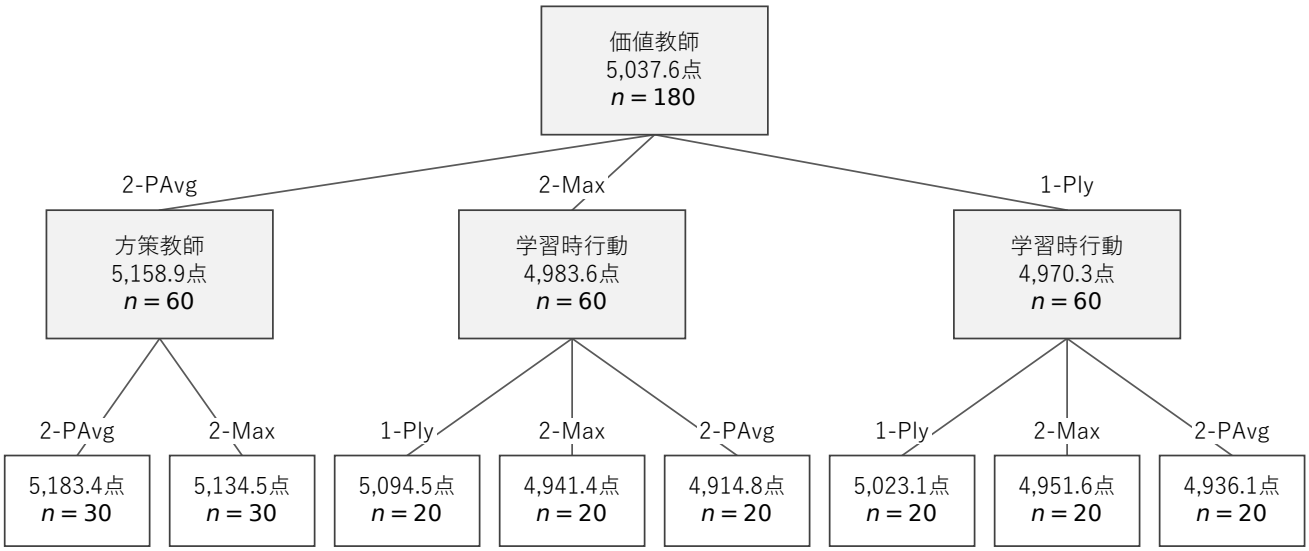


図 1 Product(3) の平均得点を目的変数とする回帰決定木（最大深さ 2）．節点内の得点は所属モデルの平均得点，n は学習条件とシードの組の数を示す．枝は平均得点の高い順に左から配置した．

表 1 学習条件の選択肢．

軸	選択肢	役割
価値教師	1-Ply / 2-Max / 2-PAvg	価値の目標値
方策教師	なし / 2-Max / 2-PAvg	方策の正解手
行動源	1-Ply / 1-Ply×π	着手用の評価値
	2-Max / 2-Max×π	
	2-PAvg / 2-PAvg×π	

表 2 価値教師別の平均得点．

価値教師	Value(3)	Policy	Product(3)
1-Ply	4979.0	4917.5	4970.3
2-Max	4956.2	4926.2	4983.6
2-PAvg	5081.9	5116.0	5158.9
全体平均	5001.9	4986.6	5037.6

Value(d) の 3 方式を評価した．各モデル・方式について，5 つの評価シードで各 2,000 ゲーム，合計 10,000 ゲームを実行した．各学習シードについて 10,000 ゲームの平均得点を求め，その後 5 学習シードに関する平均と標本標準偏差を計算した．以下では，この実験で観測した平均得点とシード間のばらつきを比較する．

5.3 学習条件と得点の関係の分析

学習条件と得点の関係を整理するため，Value(3)，Policy，Product(3) の平均得点をそれぞれ目的変数とする回帰決定木を構築した．説明変数は価値教師，方策教師，学習時行動に用いる行動価値の計算法，学習時の方策との積の有無，および学習シード[*1]とし，1 モデルの 10,000 ゲームの平均得点を 1 件として等重みで扱った．Value(3) には 200 件，Policy と Product(3) には各 180 件を用いた．学習条件はカテゴリごとに分岐し，学習シードのみ順序を仮定しないカテゴリの二分割を行う．回帰決定木の各節点では，根も含めて分割後の二乗誤差和が最小となる属性を選ぶ．

以下の分析結果を具体的に理解するための参考として，Product(3) の回帰決定木を深さ 2 まで図 1 に示す．学習条件による分岐と各条件群の得点を確認する際に参照されたい．

Value(3)，Policy，Product(3) の平均得点を目的変数とした決定木では，いずれも最初の分岐に価値教師が選ばれ，価値教師 2-PAvg の群が最も高い平均得点を得た．価値教師ごとの条件群の平均を表 2 に示す．

Value(3) では，価値教師 2-PAvg の群で次の分岐に学習時の方策との積の有無が選ばれ，行動価値そのもので着手を選ぶ 6 条件の平均は 5168.4 点であった．価値教師 2-PAvg は高得点の条件群を形成したが，その中でも経験生成時の行動選択によって得点が異なった．

Policy では，価値教師 2-PAvg の下で方策教師が次の分岐に選ばれた．価値教師 2-PAvg・方策教師 2-PAvg のときに 5148.3 点で最も高く，次いで価値教師 2-PAvg・方策教師 2-Max で 5083.8 点であった．

Product(3) では，価値教師 2-PAvg の下で方策教師が次の分岐に選ばれ，方策教師 2-PAvg の群は 5183.4 点，2-Max の群は 5134.5 点となった．一方，価値教師 2-Max と 1-Ply の下では学習時行動が選ばれ，それぞれ行動価値を 1-Ply で計算する群が 5094.5 点，5023.1 点で最も高かった．このように，価値教師によって次に得点差を説明する変数が異なった．

以上から，価値教師 2-PAvg は価値・方策のいずれを利用する場合にも有望であり，特に Policy または Product(3)

[*1] 学習シードは学習の性能差の要因に適さないが，ここでは意図的に説明変数の一つに加えた．

© 2026 Information Processing Society of Japan

表 3 両教師 2-PAvg での行動源別得点（5 学習シードの平均 ± 標本標準偏差，太字は列内最高値）．

学習時の行動源	Value(1)	Value(3)	Policy	Product(1)	Product(3)
1-Ply	**4791.5** ± 46.5	5178.7 ± 44.9	5165.8 ± 30.3	5177.2 ± 5.1	5188.5 ± 22.0
2-Max	2336.2 ± 489.8	**5187.9** ± 40.1	**5189.6** ± 22.3	**5200.5** ± 30.9	**5222.4** ± 16.6
2-PAvg	2077.0 ± 557.4	5139.8 ± 116.9	5163.0 ± 34.6	5157.4 ± 39.2	5180.4 ± 41.4
1-Ply×π	2005.8 ± 424.8	5044.1 ± 85.6	5154.4 ± 21.9	5161.9 ± 22.4	5213.2 ± 8.5
2-Max×π	1402.3 ± 491.1	4795.2 ± 264.2	5113.1 ± 28.6	5119.2 ± 23.8	5169.6 ± 13.6
2-PAvg×π	1209.0 ± 442.1	4888.5 ± 118.7	5103.6 ± 68.3	5103.2 ± 47.5	5126.0 ± 57.0

を用いる場合には，方策教師にも 2-PAvg を用いる組み合わせが高い平均得点を得た．以下では，この組み合わせに着目して経験生成方法と学習後の利用方法を調べる．

5.4 学習経験の生成と得点の関係の分析

前節の結果を踏まえ，両教師を 2-PAvg に固定し，行動源 6 通りを比較した結果を表 3 に示す．表の行は行動源，列は学習後の利用方法の比較に対応する．

まず学習後の利用方法ごとに行動源を比較する．価値をそのままプレイに用いる Value(1) では，行動源 1-Ply で 4791.5 点となり最も高く，他の 5 通りでは得点が低下した．一方，Value(3)，Policy，Product(1)，Product(3) の 4 方式は，教師とは異なる計算法で着手を選ぶ行動源 2-Max で最も高い平均得点を得た．この条件では，Policy が 5189.6 点，Product(3) が 5222.4 点となり，いずれもベースラインの Value(3)（5080.5 点）を上回った．また，学習時に方策確率と行動価値の積で着手を選ぶ（例えば 1-Ply に対する 1-Ply×π）ものは，ほとんどの比較で平均得点が下がった．

次に，表の各行で学習後の利用方法を比較すると，これら 6 条件では，深さ 1 と 3 のいずれでも Product(d) が Value(d) を上回った．

さらに，対象を全 40 条件に広げて学習後の利用方法を比較すると，Value(1) から Value(3) への探索の深化によって 200 モデルすべてでゲーム平均得点が改善した．方策を持つ 36 条件では，すべての条件で，Product(3) の平均得点は Policy を上回った．一方，Product(3) が Value(3) を上回ったのは 20 条件であり，価値探索に方策を加えた効果は学習条件によって異なった．したがって，方策に価値探索を組み合わせることは広い条件で得点向上につながったが，方策を加えれば常に価値探索を上回るという結果ではなかった．

5.5 各プレイヤ自身の訪問盤面による分析

得点の比較に続いて，プレイヤ自身が訪れた局面と実際の着手を分析する．まず，学習時の行動源を 1-Ply に揃えたベースラインと両教師 2-PAvg の条件を，ともに Value(3) で利用して学習法の違いを比較する．1 プレイヤ当たり計 2,500 ゲームのプレイ結果から，最善手一致率 [7] と一手損失 [14] を比較する．一手損失は完全解析値に基づく最善手から選択手への期待得点の減少を非負値で表し，最善手一致率の計算では一手損失が 10^{-6} 以下の手をすべて正解とする．盤上のタイル値の総和の半分を進行度とし，グラフの横軸に用いる．図 2 は，両教師 2-PAvg・行動源 1-Ply の指標からベースラインの指標を引いた差を示す．上は最善手一致率差（pp），下は一手損失差（点）である．

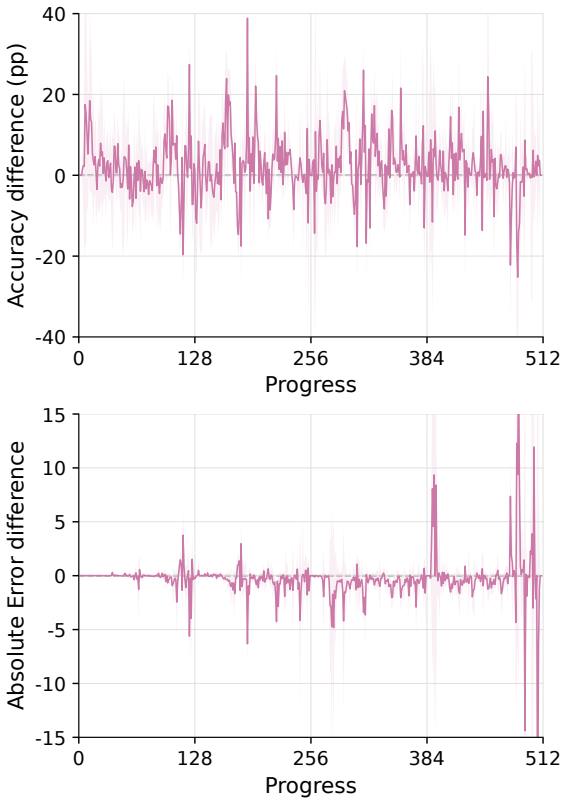


図 2 両教師 2-PAvg・行動源 1-Ply の Value(3) の指標からベースラインの Value(3) の指標を引いた差．

両教師 2-PAvg・行動源 1-Ply では，一致率差が正，一手損失差が負となる進行度が多く見られた．得点の改善とともに，自身の訪問盤面での判断に改善が見られる．

次に，学習条件を両教師 2-PAvg・行動源 2-Max に固定し，同じモデルの利用方法を比較する．対象は Value(3)，Policy，Product(1) とし，先の比較と同様に 1 プレイヤ当たり計 2,500 ゲームのプレイ結果から，最善手一致率と一手損失を求める．

ここでは，ゲーム平均得点が近い Policy と Value(3) に着目し，その比較を図 3 に示す．Policy から Value(3) を

© 2026 Information Processing Society of Japan

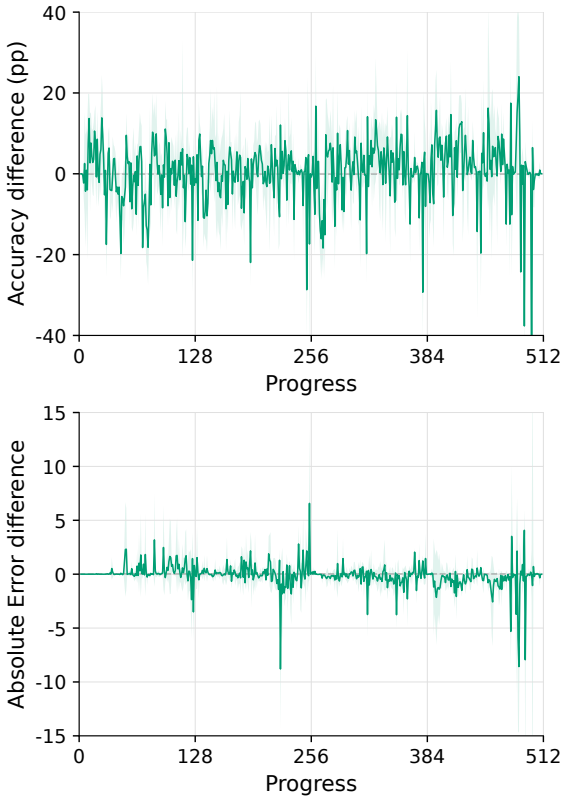


図 3 両教師 2-PAvg・行動源 2-Max の Policy の指標から Value(3) の指標を引いた差.

引いた差であり，上は最善手一致率差，下は一手損失差である．

Policy と Value(3) の最善手一致率および一手損失の差は，進行度によって正負に変動した．ゲーム平均得点が近い二つの利用方法でも，実際のプレイ中の判断には進行段階ごとの差があった．しかしながら，ゲーム局面の特性とこれらの差との関係について，明らかな傾向を見出すことはできなかった．

5.6 自身の訪問盤面における探索評価値

前節の一致率と損失に続いて，実際の訪問盤面で探索が返す評価値の正確さを調べる．図 4 は，ベースライン，両教師 2-PAvg で行動源を 1-Ply としたもの，および，両教師 2-PAvg で行動源を 2-Max としたものの計 3 条件の Value(3) について，各条件のプレイヤ自身が訪れた盤面で真の行動価値と深さ 3 探索の行動価値を対応させたものである．

ベースラインは全体として $y = x$ に近いが，初期盤面付近で過大評価を起こしていることが確認できる．これは著者らの既存研究 [15] と一致する．両教師 2-PAvg・行動源 1-Ply では，評価値が完全解析値より低い側へ大きくずれた．行動源 2-Max では，完全解析値よりやや低いものの，ベースラインの初期盤面付近の過大評価や行動源 1-Ply の大きな過小評価が抑えられ，おおむね直線的に対応する点で望ましい分布と考えられる．両教師 2-PAvg であるこれら 2 条件の Value(3) は，ベースラインより平均得点が高かった．

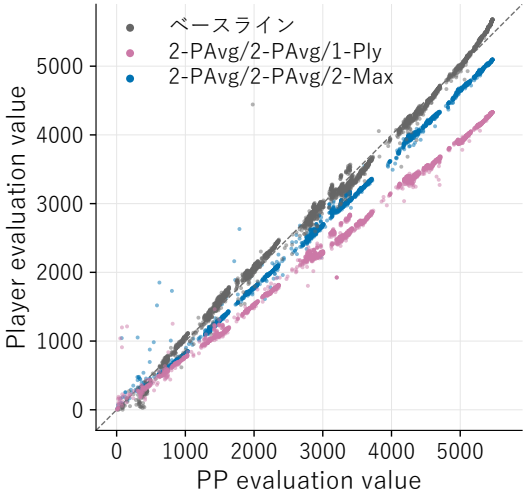


図 4 各条件のプレイヤ自身が訪れた盤面における完全解析の最適手の行動価値．横軸は完全解析値，縦軸は Value(3) の評価値.

6. 議論

6.1 学習目標の選定：方策を教師計算に用いる意義

第 5.3 節では，価値教師 2-PAvg の群が，Value(3)，Policy，Product(3) のいずれでも高い平均得点を得た．方策を利用する二つの方式では，方策教師も 2-PAvg とする群の平均得点が高かった．このことから，方策は学習後の着手選択に使うだけでなく，探索木から教師を計算する際にも利用する価値があると考えられる．

2-PAvg では，価値教師の継続部分には方策確率で重み付けした行動価値を用い，方策教師にはその計算法に基づく行動価値が最大の手を用いる．両者を 2-PAvg とすることで，高得点の Policy と Product が得られた．ただし，Value(3) の平均得点を目的変数とする決定木では，価値教師の次に学習時の行動合成が選ばれており，両教師 2-PAvg がすべての利用方法・行動源で最良であるとまではいえない．

6.2 学習経験の生成：教師と着手の計算法の使い分け

教師を 2-PAvg に揃えた第 5.4 節の比較では，着手を 2-Max で選ぶ組み合わせが, Policy, Product, および Value(3) で高い平均得点を得た．教師計算では方策重み付き平均を用い，経験生成では最大値方式の探索を用いるという使い分けが有効な構成であった．

一方，Value(1) で利用する場合には，行動源 1-Ply が最も高い平均得点を得た．学習中に深い探索で手を選ぶことは，学習後に浅い価値評価で良い手を選べることを保証しない．したがって，学習時の行動源は，最終的に Value(1)，深い価値探索，Policy のどれを利用するかも考慮して選ぶ必要がある．

© 2026 Information Processing Society of Japan

また，両教師 2-PAvg の比較では，学習時に方策と行動価値の積で着手を選ぶと，15 比較中 14 比較で行動価値そのもので着手を選ぶ条件より平均得点が低かった．一方，学習後の Product(3) は Policy より高い平均得点を得た．このように，方策との積を用いる効果は，経験を生成する段階と学習結果を利用する段階で異なった．

6.3 学習結果の利用：得点と行動選択

学習後の利用方法について，第 5.4 節では，価値に基づくプレイの探索を深くすると全 200 モデルで得点が改善した．また，両教師 2-PAvg・行動源 2-Max では，Policy 単独が Value(3) に近い平均得点を得た．探索結果を教師として学習した方策を，プレイ時には方策推論だけで利用できることは，この構成の利点である．

方策に価値探索を組み合わせた Product(3) の平均得点は，36 条件すべてで Policy を上回った．一方，Value(3) との優劣は条件によって異なった．

6.4 評価値の正確さと行動選択を分けた解釈

第 5.6 節の散布図では，両教師 2-PAvg で学習した Value(3) の評価値が，自身の訪問盤面でも完全解析値より低い側へずれた．一方で，ベースラインよりゲーム得点が高かった．教師計算法が異なることも踏まえると，評価値の絶対水準だけで学習結果の有効性を判定することはできない．このうち，両教師 2-PAvg・行動源 2-Max は，教師計算と学習時の着手選択で計算法を使い分ける構成である．推測に過ぎないが，Double Q-learning[16] のように，学習時の行動選択と評価値の計算を分けることで，過大評価を抑えつつ，より良い手を選べるようになった可能性がある．

6.5 実行時間での再評価

ここでは，学習時間と学習後の行動選択時間について追加で検討する．

6.5.1 学習時間を考慮した比較

主実験では 250,000 学習ステップに揃えたが，所要時間の 5 シード平均は，ベースラインが 5.40 ± 0.08 時間，両教師 2-PAvg・行動源 2-Max が 27.87 ± 0.51 時間であった．そこで，ベースラインを新たに 5 シードで初期化し，終了条件のみを 27 時間に変更して学習した．学習ステップ数は約 107 万〜125 万で，各最終モデルを主実験と同じ評価シードの 10,000 ゲームで評価した．

27 時間学習したベースラインの平均得点は，Value(1) が 4157.6 点，Value(2) が 4768.1 点，Value(3) が 4983.3 点であり，250,000 学習ステップでの 4358.2 点，4923.7 点，5080.5 点をいずれも下回った．この補足実験では，ベースラインの学習時間を延ばしても，両教師 2-PAvg・行動源 2-Max との得点差は解消されなかった．この結果から，今回の条件では，学習目標と経験生成方法の選択が性能に関係することが示唆された．

表 4 共通盤面での行動選択時間（ms/盤面，5 学習シードの平均 ± 標本標準偏差）．

方式	単盤面	256 並列
Product(1)	7.15 ± 0.248	0.032 ± 0.0013
Policy	5.19 ± 0.186	0.022 ± 0.0009
Value(3)	14.30 ± 0.593	0.800 ± 0.0261

6.5.2 行動選択時間を考慮した比較

次に，両教師 2-PAvg・行動源 2-Max の 5 学習シードの最終モデルを用い，共通の 2,048 盤面で Product(1)，Policy，Value(3) の行動選択時間を単盤面・256 並列で比較した．256 並列の時間は 1 盤面当たりに換算した時間である．

表 4 に 3 方式の行動選択時間を示す．全 5 シード・両バッチ条件で，行動選択時間が短い順に，Policy，Product(1)，Value(3) であった．Value(3) の時間は Policy に対して単盤面で 2.76 倍，256 並列で 36.28 倍，Product(1) に対してそれぞれ 2.00 倍，25.03 倍であった．

両教師 2-PAvg・行動源 2-Max では，Policy と Value(3) の平均得点はそれぞれ 5189.6 点と 5187.9 点で近く，共通盤面での行動選択時間は Policy の方が短かった．Policy は根盤面の方策推論で着手を選ぶため，Value(3) で必要となる確率分岐の展開と葉の価値評価を省ける．探索結果から学習した方策を単独で利用することで，深い価値探索に近い平均得点と，短い行動選択時間を両立できた．

Product(1) は Policy に合法手ごとの afterstate 価値評価を加え，平均 5200.5 点を得た．行動選択時間は Policy に対して，単盤面で 1.38 倍，256 並列で 1.45 倍であり，Value(3) より短かった．平均得点の差は小さいものの，深さ 1 の価値評価との積を用いることで，方策単独からの時間増加を抑えながらスコアを改善できた．この条件では，行動選択時間の短さを重視する場合は Policy，浅い価値評価を加える場合は Product(1) が利用候補となる．

6.6 先行研究の報告値との比較

最後に，ミニ 2048 を対象とする Matsuzaki と Terauchi[17] の報告値と比較する．先行研究の 3-ply は本研究の深さ 3 に対応する．主な違いは，先行研究が N タプルネットワークで価値を TD 学習するのに対し，本研究は深さ 2 の探索結果から CNN の方策と価値を学習する点である．

1 回の学習量は，先行研究が 5 億行動，本研究が 256 並列で 250,000 学習ステップ，約 6400 万回のゲーム処理（終端の初期化を含む）である．本研究では学習時にも探索を行うため，処理回数の比は計算時間の比とは異なる．

先行研究の 3-ply で最も高い平均得点は 5117.8 点であった．本研究の両教師 2-PAvg・行動源 2-Max の条件は，Pol-

© 2026 Information Processing Society of Japan

icy で 5189.6 点，Product(3) で 5222.4 点を得た．いずれも各モデルを 10,000 ゲームで評価した値で，先行研究は 10 回，本研究は 5 回の学習で得たモデルの評価得点を平均している．探索を使わない方策単独でも，先行研究の 3-ply の平均得点を上回った．

先行研究が選抜したモデルの平均得点は，3-ply で 5175.9 点, 7-ply で 5388.0 点である．評価はそれぞれ 10,000, 1,000 ゲームで，複数回の学習の平均とは異なる．本研究で最も高い平均得点である 5222.4 点は 3-ply の報告値を上回ったが，7-ply には届かなかった．

7. まとめ

本研究では，2048 で学習した方策を使うことが性能改善につながるかを調べるため，深さ 2 の Expectimax 探索から方策と価値を学習する AlphaZero 型の学習を構成した．ミニ 2048 で 40 条件を各 5 シードで学習し，学習目標の選定，学習経験の生成，学習結果の利用の三つの観点から比較した．

学習目標の選定では，価値教師 2-PAvg の条件群が Value(3)，Policy，Product(3) で高い平均得点を得た．特に Policy と Product(3) では，方策教師も 2-PAvg とする群の平均得点が高かった．

学習経験の生成では，両教師 2-PAvg の下で，Value(1) には行動源 1-Ply，Value(3)，Policy，Product(1)，Product(3) には行動源 2-Max が最も高い平均得点を与えた．教師と学習時の着手で計算法を使い分ける構成が有効であり，経験生成方法は学習後の利用方法も考慮して選ぶ必要があった．

学習結果の利用では，Value(1) から Value(3) への探索の深化で全 200 モデルの得点が改善した．両教師 2-PAvg・行動源 2-Max では，Policy 単独で 5189.6 点，Product(3) で 5222.4 点となり，ベースラインの Value(3) の 5080.5 点を上回った．同じ学習条件の Value(3) は 5187.9 点であり，Policy 単独でもこれに近い平均得点を得た．全 36 条件で Product(3) の平均は Policy を上回ったが，Value(3) との優劣は学習条件によって異なった．

また，各プレイヤ自身の訪問盤面を分析した結果，平均得点が近い Policy と Value(3) でも，進行段階ごとの判断に差が見られた．今後は，盤面を抽出して個々の盤面における行動選択の傾向を検証し，判断の差が生じた要因の分析を進める．さらに，4×4 盤面の 2048 にも今回の手法を導入し，方策を学習目標に用いることが性能改善につながるかを調べる．

参考文献

[1] M. Szubert and W. Jaśkowski: Temporal Difference Learning of N-Tuple Networks for the Game 2048, IEEE Conference on Computational Intelligence and Games, pp.1–8 (2014).

[2] H. Guei, L.-P. Chen and I-C. Wu: Optimistic Temporal Difference Learning for 2048, *IEEE Transactions on Games*, Vol.14, No.3, pp.478–487 (2022).

[3] K. Matsuzaki: Developing Value Networks for Game 2048 with Reinforcement Learning, *Journal of Information Processing*, Vol.29, pp.336–346 (2021).

[4] D. Silver et al.: A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play, *Science*, Vol.362, No.6419, pp.1140–1144 (2018).

[5] I. Antonoglou, J. Schrittwieser, S. Ozair, T. K. Hubert and D. Silver: Planning in Stochastic Environments with a Learned Model, *ICLR 2022* (2022).

[6] C.-Y. Kao, H. Guei, T.-R. Wu and I-C. Wu: Gumbel MuZero for the Game of 2048, *2022 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pp.42–47 (2022).

[7] 山下修平, 金子知適, 中屋敷太一：3 × 3 盤面の 2048 の完全解析と強化学習の研究, ゲームプログラミングワークショップ 2022 論文集, pp.1–8 (2022).

[8] I-C. Wu, K.-H. Yeh, C.-C. Liang, C.-C. Chang and H. Chiang: Multi-Stage Temporal Difference Learning for 2048, *Technologies and Applications of Artificial Intelligence*, LNCS Vol.8916, pp.366–378 (2014).

[9] K.-H. Yeh, I-C. Wu, C.-H. Hsueh, C.-C. Chang, C.-C. Liang and H. Chiang: Multistage Temporal Difference Learning for 2048-Like Games, *IEEE Transactions on Computational Intelligence and AI in Games*, Vol.9, No.4, pp.369–380 (2017).

[10] K. Matsuzaki: Systematic Selection of N-Tuple Networks with Consideration of Interinfluence for Game 2048, *2016 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pp.186–193 (2016).

[11] W. Jaśkowski: Mastering 2048 With Delayed Temporal Coherence Learning, Multistage Weight Promotion, Redundant Encoding, and Carousel Shaping, *IEEE Transactions on Games*, Vol.10, No.1, pp.3–14 (2018).

[12] 渡邊翔太: ゲーム「2048」への Monte-Carlo Softmax 探索の適用, 高知工科大学大学院修士学位論文 (2024).

[13] W. Wang: Enhancing Data Utilization in Game AI: Deep Reinforcement Learning Strategies for the Game 2048, Ph.D. dissertation, Kochi University of Technology (2025).

[14] S. Terauchi, T. Kubota and K. Matsuzaki: Using Strongly Solved Mini2048 to Analyze Players with N-tuple Networks, *Technologies and Applications of Artificial Intelligence (TAAI 2023)*, CCIS Vol.2075, pp.165–178 (2024).

[15] 金子智弥, 松崎公紀: ミニ 2048 における Double Q 学習の効果 ―過大評価を乗り越えるために―, ゲームプログラミングワークショップ 2025 論文集, pp.179–186 (2025).

[16] H. van Hasselt: Double Q-learning, *Advances in Neural Information Processing Systems*, Vol.23, pp.2613–2621 (2010).

[17] K. Matsuzaki and S. Terauchi: Yet More Optimistic Temporal Difference Learning for Game Mini2048, *2024 IEEE Conference on Games (CoG)*, pp.1–8 (2024).

© 2026 Information Processing Society of Japan