

論文題目

# connect4を題材とした強化学習AIの 判断の可視化

Visualization of Reinforcement Learning AI Decision  
Using Connect4 as an Example

指導教授

萩原 将文 教授

慶應義塾大学 理工学部 情報工学科

令和5年度

学籍番号 62019277

村上 花恋

# 目 次

あらまし	1
第1章 はじめに	2
第2章 関連研究	4
2.1 強化学習 . . . . .	4
2.1.1 ボードゲームへの応用 . . . . .	5
2.1.2 AlphaZero . . . . .	7
2.1.3 ニューラルネットワーク . . . . .	8
2.1.4 PV-モンテカルロ木探索 . . . . .	8
2.1.5 ボードゲーム AI の問題点 . . . . .	9
2.2 XAI . . . . .	10
2.2.1 概要 . . . . .	10
2.2.2 ゲームにおける XAI . . . . .	11
2.2.3 contrastive explanation . . . . .	11
2.2.4 importance . . . . .	13
2.2.5 ボードゲーム学習支援 . . . . .	14
第3章 提案手法の概要	15
3.1 提案手法のアルゴリズム . . . . .	15
3.2 importance . . . . .	20
3.3 connect4 . . . . .	21
3.4 alphazero_baseline . . . . .	22
3.5 提案手法の connect4 への適用 . . . . .	22

<b>第4章 評価実験</b>	<b>25</b>
4.1 データ実験 . . . . .	25
4.1.1 データセット . . . . .	25
4.1.2 データ実験の評価指標 . . . . .	26
4.1.3 group prediction rate . . . . .	28
4.1.4 stone prediction rate . . . . .	28
4.1.5 比較手法 . . . . .	29
4.1.6 データ実験における提案指標の計測 . . . . .	31
4.1.7 実験結果(データ実験) . . . . .	31
4.1.8 考察(データ実験) . . . . .	32
4.2 インタフェース実験 . . . . .	33
4.2.1 実験手順 . . . . .	33
4.2.2 インタフェース実験の評価指標 . . . . .	34
4.2.3 実験結果(インターフェース実験) . . . . .	36
4.2.4 考察(インターフェース実験) . . . . .	38
<b>第5章 結論</b>	<b>41</b>
<b>謝辞</b>	<b>43</b>
<b>参考文献</b>	<b>44</b>
<b>付録</b>	<b>50</b>
<b>付録A データ実験の詳細</b>	<b>50</b>
A.1 使用したモデルの詳細 . . . . .	50
A.2 対戦結果の詳細 . . . . .	50
A.3 データ実験に使用したモデルの詳細 . . . . .	51
A.4 末尾のグループ化 . . . . .	52
A.5 グレーボックス的手法 . . . . .	53

<b>付録 B 各種アルゴリズムの詳細</b>	<b>55</b>
B.1 PV-モンテカルロ木探索	55
B.2 提案手法のアルゴリズム	55
B.3 比較手法のアルゴリズム	55
B.4 提案手法におけるニューロ補間	55
B.5 インタフェース実験における提案手法の変更	61
<b>付録 C alphazero_baseline</b>	<b>65</b>
C.1 ニューラルネットワーク	65
C.2 alphazero_baseline のパラメータ更新	67
<b>付録 D インタフェース実験の詳細</b>	<b>69</b>
D.1 被験者のデータ	69
D.2 第1段階	70
D.2.1 振り返りモード (GUI) の詳細	71
D.2.2 1日目	72
D.2.3 2日目	72
D.3 第2段階	74
D.4 質問項目の詳細	75
D.5 結果データ	75
<b>付録 E 追加実験の詳細</b>	<b>77</b>
E.1 被験者のデータ	77
E.2 実験設定の詳細	77
E.2.1 振り返りモード (GUI) の詳細	78
E.2.2 アルゴリズムの変更	79
E.3 質問項目の詳細	81

# あらまし

2024年現在，チェス1，囲碁，将棋などのメジャーな2人用ボードゲームにおいてAIは人間を遙かに凌駕するようになった[1][2][3]。しかし，AlphaZero[4]の登場から5年が経過した今もなおその十分な説明手法は登場していない。そこで，AIの予測する進行図を複数提示することでAIの判断根拠の可視化を試みた。題材として対戦型ボードゲームの1つであるconnect4[5]を選択した。実験はAI同士の対戦データを用いたデータ実験と被験者からのデータを用いたインターフェース実験の2種類を行った。結果としてデータ実験では提案手法が比較手法よりも高い予測精度を示し，インターフェース実験では提示する情報量が極端に大きいまたは小さい場合に，提案手法は比較手法よりも被験者が「AIの意図を掴めた」と感じやすいという結果が出た。

# 第1章

## はじめに

近年の AI 分野の発展は目覚ましく、画像分類などの単純なルールで記述する事が困難なタスクや、更には長らく人間に固有の技術であると考えられてきた画像や文章の生成の分野においてさえ、高い性能を発揮している [6]。特に直近 3 年の Stable Diffusion[7]、Instruct GPT[8] の登場により、専門家でない人々が AI を使用する機会が増加した。そのため、人間と AI の将来的な関係性に対する考察の必要性が高まっているといえる。この「優れた AI に対して人間はどう接すべきか」という命題を考える際には、既にスピードや能力において人間を大きく凌駕した AI を題材とし、手法の構築や実験を行うのが適当である。そのため、囲碁、将棋、チェスなどの主要なボードゲームにおいて人間の実力を凌駕したパフォーマンスを誇る AlphaZero を題材に、説明性付与を試みる。

優れた AI が社会で広く実用化され受容されるためには、AI の出力が生み出される過程の透明性、信頼性の担保が必要不可欠である。説明性が必要とされるのは医療や金融などの判断の慎重さが求められるべき分野のみではない。上述の様に 2024 年現在、画像生成、文章生成 AI はオープンソース化され、広く普及した。そのような状況においても、G7 広島サミットでは国際的な協調の観点から AI の透明性を確保する重要性が指摘されている [9]。また、日本国内においても「信頼できる AI」を実現する必要性が認識されている [10]。それに加えて、将来的には「優れた AI に学ぶ」という探求心や学習意欲からも説明性へのニーズが広く湧き起こる事が予測される。現に囲碁、将棋、チェスなどの主要なボードゲームをオンラインでプレイできるサービスには、ゲーム終了後の振り返り（感想戦）において AI の判断や AI による進行図を閲覧できるサービス

が数多く提供されている [11][12] . 研究においてもボードゲームの学習支援という形で「AIに学ぶ」試みが存在する . しかし , 後述するようにそれらの研究は「王手」などの各ゲームに特有の知識(ドメイン知識)に依存するものが多い .

そこでなるべくドメイン知識を用いず , AI の判断を可視化する手法を考案しその有効性を検証する .

続く第 2 章では関連研究について記載し , 第 3 章では提案手法の詳細を述べ , 第 4 章に行った 2 つの実験の結果と考察を記す .

## 第2章

### 関連研究

この章ではまず，既存のボードゲーム AI について AlphaZero を中心に強化学習的枠組みからその理論を説明する。次に AlphaZero の問題点について述べ，それを補完する既存手法，その課題について述べる。

#### 2.1 強化学習

強化学習はタスクを選択をする主体と環境のやり取りとして定式化し，その相互作用から学習を行う形でタスクに取り組む分野である [13]。

強化学習タスクでは能動的に行動を行う主体と，主体が働きかける環境を設定する。図 2.1 に示すように環境の状況は状態  $s$  として定義され， $s$  に対する主体の行動  $a$  によって環境の次の状態  $s' (= T(s, a), T$  は遷移関数) と主体が獲得する報酬  $r$  が決定されると仮定する。その仮定の下，環境から主体に与えられる報酬の合計  $G (= \sum r$  以下収益と記載) を最大化することを目標とする。報酬を大きくするために各状態  $s$  に応じて適切な(より大きな報酬をもらえる可能性が高い) 行動を選択する必要がある。そのため状態  $s$  に対して行動  $a$  をとった場合の収益に対して見積もりをとり，見込まれる値が最も大きい行動を選択することでより大きな収益を獲得できると期待できる。このような収益の見積もりを  $Q(s, a)$  とした場合，

$$a_m = \operatorname{argmax}_a Q(s, a) \quad (2.1)$$

となる  $a_m$  を選択することによって収益の最大化が期待される。また，状態  $s$  から獲得できる収益の合計の予想値  $V(s)$  (以下状態価値関数と表記) は，最適な

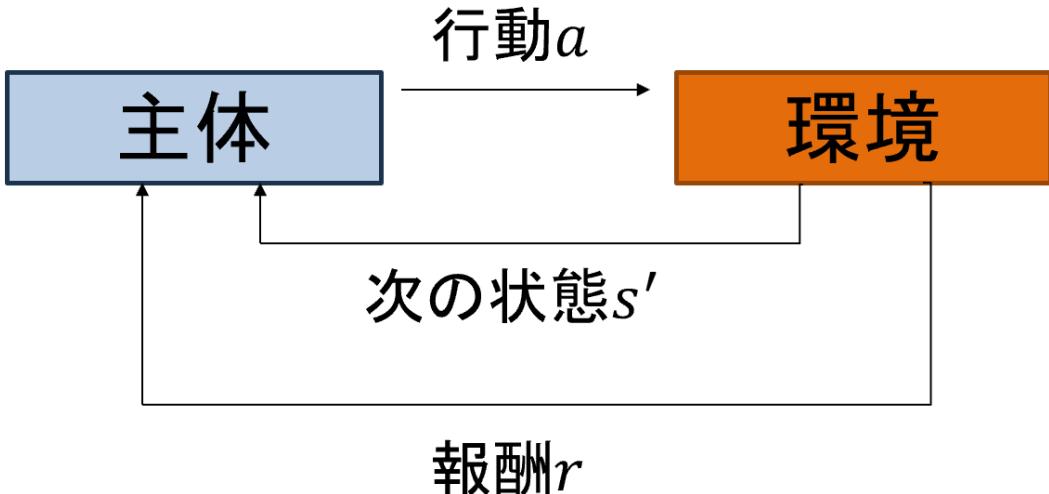


図 2.1: 強化学習

行動  $a_m$  を取った場合の値として推定される .

$$V(s) = Q(s, a_m) \quad (a_m = \operatorname{argmax}_a Q(s, a)) \quad (2.2)$$

強化学習手法によってタスクの最適化を図る際にはこの  $V(s), Q(s, a)$  を正しく推定することが直接的な目標となる .  $V(s), Q(s, a)$  は主体が実際に環境とやり取りを行う（タスクを実行していく）中で改善されていくことが期待される . Temporary Difference 法や Monte Carlo 法等が基本的な  $V(s), Q(s, a)$  の更新則であり , DQN[14] や Rainbow[15] 等はニューラルネットワークを使用して  $V(s), Q(s, a)$  を推定することでより高い性能を発揮している .

### 2.1.1 ボードゲームへの応用

ボードゲームでは通例 , 状態  $s$  は盤面の状況 , 行動  $a$  はプレイヤーの選択 , 報酬  $r$  はゲームの最後に勝敗として与えられる . 図 2.2 に状態  $s$ (ゲームの状況) と行動  $a$ (プレイヤーの選択) によって盤面は次の状態  $s'$  に遷移し , 次の行動  $a'$ (他のプレイヤーによる選択) を受け付ける , というサイクルとしてゲームの進行を定式化して表現することができる . また , 上述した強化学習における状態価値関数  $V(s)$  の推定は「ある盤面はプレイヤーにとって勝利に近いのか」を表現し

ていると解釈され，行動価値関数  $Q(s, a)$  は「ある盤面である行動をした場合のプレイヤーの勝率」を表していると解釈される。AlphaGo[16] や StockFish[17]，DeepLearningShogi[18] では状態  $s$  は最新  $N$  ステップの盤面と手数などのゲームのプレイにおいて重要な情報である。状態は行列等の形式に抽象化され，行動は次にプレイヤーが打つ箇所の座標となる。

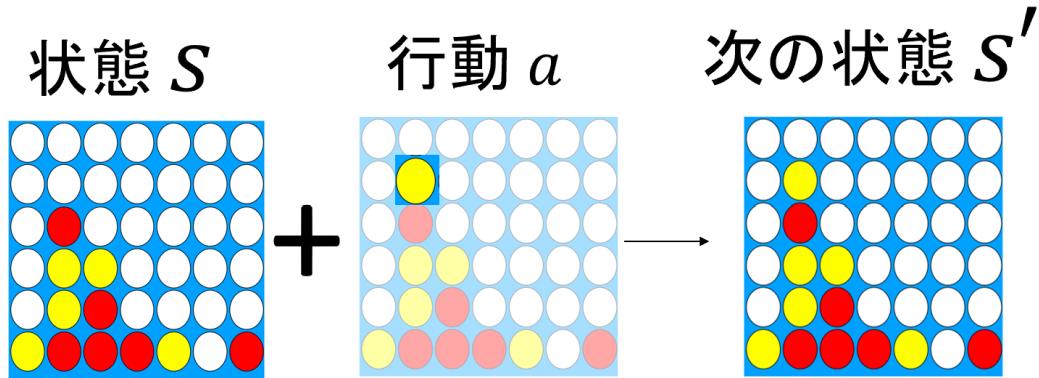


図 2.2: ボードゲームにおける強化学習モデル

また，図 2.3 に示すように本論文で使用した alphazero\_baseline モデル [19] における入力は最新の盤面の状態を空白を 0，先番(赤)の石の位置を 1，後番(黄)の位置を -1 として抽象化した  $6 \times 7$  の行列となる。また，後述する connect4 の

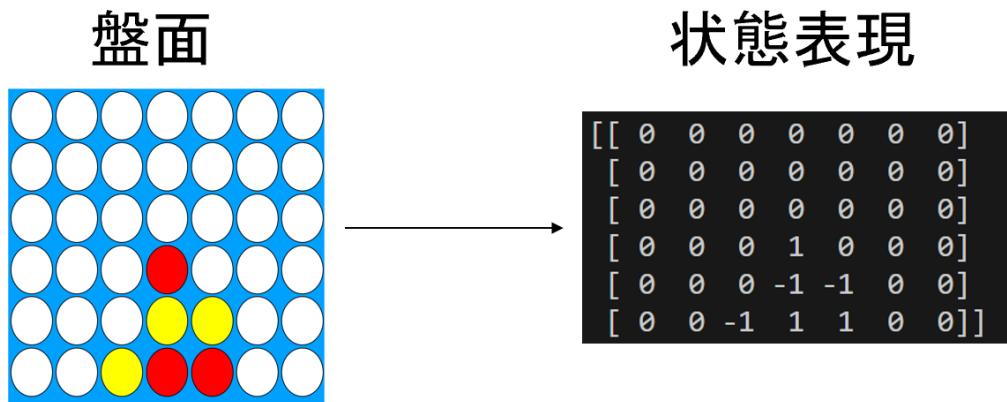


図 2.3: connect4 盤面の状態表現

ルール上の制約により次にプレイヤーが打つ箇所(行動)は列の数と同数の 7 種

類に限定される。

### 2.1.2 AlphaZero

AlphaZero は 2016 年に登場し、囲碁の元世界チャンピオンであるイ・セドルに対して 4 勝一敗の成績を収めた AlphaGo の汎用版であり、囲碁、将棋、チェスなどの主要なボードゲームにおいて人間の実力を凌駕したパフォーマンスを誇る。AlphaZero は先述の  $V(s), Q(s, a)$  を推定する際にニューラルネットワークと PV-モンテカルロ木探索アルゴリズムと呼ばれるモンテカルロ木探索アルゴリズムの変種を使用する。

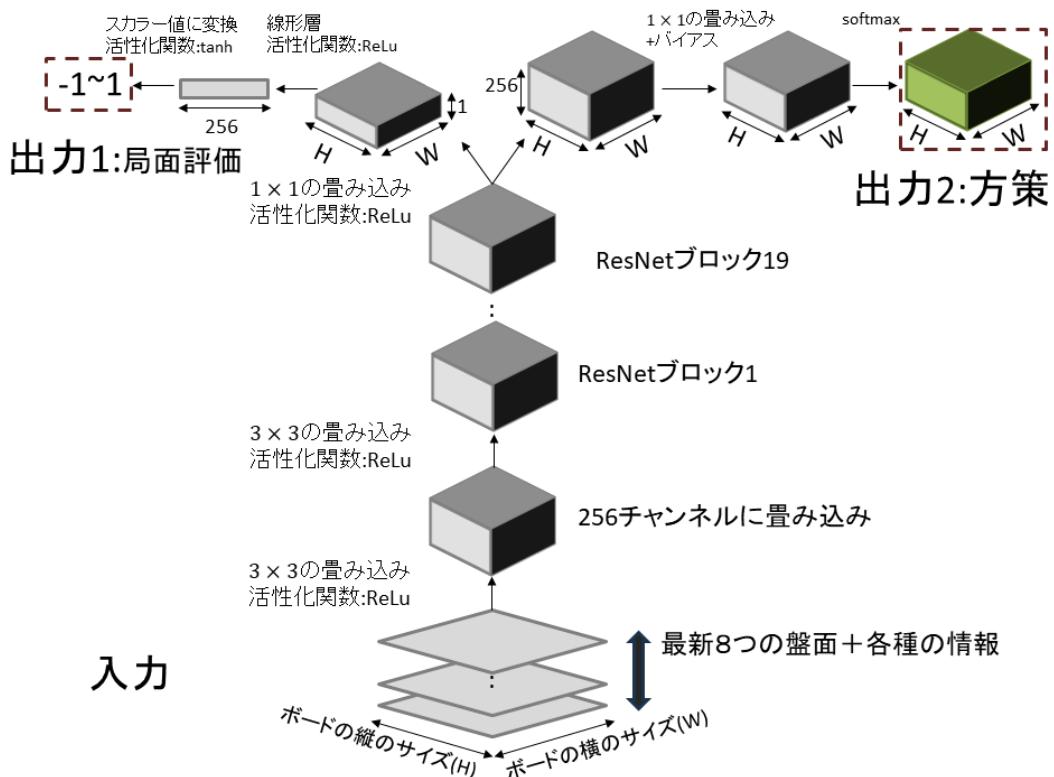


図 2.4: AlphaZero ネットワークの構造

### 2.1.3 ニューラルネットワーク

AlphaZero におけるニューラルネットワークの構造を図 2.4 に示す。AlphaZero 内のニューラルネットワークに対する入力は最新 8 ステップの盤面 ( $\{s_{-7}, \dots, s_0\}$ ,  $s_{-i}$  は  $i$  ステップ前の盤面,  $s_0$  は現在の盤面) であり, 出力は方策  $P(\{s_{-7}, \dots, s_0\})$  と 局面評価  $V(\{s_{-7}, \dots, s_0\})$  の 2 種類である。ネットワークは 1 つの畳み込み層と 20 の残差結合ネットワークで構成されている。

方策は「現在の状況から次にどこを選択すべきか」を表現しており, 次に選択すべき座標を確率分布の形式で表現する。方策内の値の大きさが AI によるその着手の評価と解釈され, 成分が大きい座標を次に選択することが推奨される。また, 局面評価  $V(\{s_{-7}, \dots, s_0\})$  は「(過去 7 ステップ分の情報を含めた) 現在の状況  $s_0$  は勝利に近いのか」を表現しており, 値が上限に近ければ近い程, 現在の状況  $s_0$  が次の着手を選択するプレイヤーにとっての勝利に近いことを表している。

### 2.1.4 PV-モンテカルロ木探索

PV-モンテカルロ木探索ではニューラルネットワークから得た方策  $P(s)$ (以下  $s = \{s_{-N+1}, \dots, s_0\}$  と表記する) と局面評価  $V(s)$  をシミュレーションによって改善する。PV-モンテカルロ木探索ではシミュレーションによって  $s$  または  $(s, a)$ (状態と行動の組) をノード, 各行動  $a$  を枝とした決定木を構築する。最終的に各ノード  $s$  から派生する各行動  $\{a_1, a_2, \dots, a_n\}$  の探索回数の分布  $N(s)$ (=  $N(s, a_1), N(s, a_2), \dots, N(s, a_n)$ ) が改善された方策となる。一方, 局面評価もまたモンテカルロ木探索により決定木が拡張されるなかで更新される。付録 B の Algorithm1, Algorithm2 に決定木と局面評価  $V(s)$  の詳細な更新アルゴリズムを示し, ここでは大まかな流れを示す。PV-MCTS のアルゴリズムの流れは以下の通りである。

1. まず, 探索の開始地点となるノード  $s$  を決定する。
2. 再帰部分では以下の処理を再帰的に呼び出す。

- (a) ノード  $s$  を探索したことがない場合ニューラルネットワークから出力された方策  $P(s)$  と局面評価  $V(s)$  を返却する .
- (b) ノード  $s$  を探索したことがある場合以下の puct スコア  $U(s, a)$  と行動価値関数  $Q(s, a)$  の和が最大となる子ノード  $s_c (= T(s, a))$  を選び,  $s_c$  に対してさらに再帰的に探索を行いその結果である  $P(s_e), V(s_e)$  ( $s_e$  は再帰処理の結果たどり着く決定木の端のノード) を返却する . ( $N(s), N(s, a)$  はそれぞれ  $s, (s, a)$  に対して探索を行った回数,  $(C_{\text{base}}, C_{\text{init}})$  はハイパラメータ))

$$U(s, a) = C(s)P(s, a) \frac{\sqrt{N(s)}}{1 + N(s, a)} \quad (2.3)$$

$$C(s) = \log \frac{1 + N(s) + C_{\text{base}}}{C_{\text{base}}} + C_{\text{init}} \quad (2.4)$$

このようなプロセスによって決定木を構築しつつ，モデルは対戦を行う .

### 2.1.5 ボードゲーム AI の問題点

AlphaZero は当時の世界最高峰の AI であるチェスの StockFish[20] や将棋の elmo[21] を凌ぐ性能を発揮した . それらの従来の有力なボードゲーム AI は設計がそのゲーム固有の知識（ドメイン知識）に依存する割合が多かったため「ある条件を満たすときにある選択をする」と言ったようにその挙動をルールの集合として解釈できる余地があった . その意味では StockFISH12 や elmo はその設計が説明性を有していたといえる . 一方で , AlphaZero のニューラルネットワークと木探索を組み合わせた手法では方策と局面評価の根拠を得ることができない . つまり AlphaZero の問題点として説明性の欠如が挙げられる . 2024 年現在 , StockFish , elmo の両者も人間の知識データへの依存を減じる方向に改良を続けており , 性能と設計の説明性がトレードオフ関係にあると言える [21][22] . 説明性の欠如は AI の判断に対する責任の不在を意味し , システムのよりハイレベル , ハイリスクなタスクへの実用化に対する妨げとなっている . そのため , 設

計段階では説明性を持たないシステムに対して説明性を付加する手法を構築する必要がある。

## 2.2 XAI

### 2.2.1 概要

XAI とは Explainable AI(説明可能 AI) の略語であり、AI モデルの結果を説明し、理由付けを行う際のシステムとユーザーのコミュニケーションに焦点当たる分野である [23]。本研究は AI の判断根拠として先読みを示す面で XAI の分野に属する研究であると言える。ここで「説明可能性」の語は「人間に理解できる形での説明を与える能力」[24] と定義され、

- いつ
- どのような
- どのように

説明を与えるかによってさらに細かく分類される。

「いつ」、つまりどの時点で説明を与えるかに関しては既存のネットワークに対して新たに説明を加える「事後的」説明と初めから動作の根拠を示せるようにネットワークやシステムを構築する「事前の」説明に分類できる [24]。

「どのような」、つまり説明の内容については「大域的」説明と「局所的」説明の 2 つに分類される。「大域的」説明は行動  $a$  を選択する主体（モデル）の全体的な方針について述べるものである一方で、「局所的」説明は主体（モデル）の個々の判断について説明する [25]。また、「どのように」、つまり説明を表現する形態としては saliency map[26]、Grad-CAM[27] といった視覚的な可視化や、教師データと予測結果との因果関係の数値的な定量化 [28]、入力の出力への寄与を表すグラフや説明文の生成 [29] が挙げられる。

本論文において構築するシステムは「事後的」「局所的」「視覚的」説明を提供する。

また，本論文は XAI 分野の中でも特に強化学習方面に対して説明を加える領域を XRL(Explainable Reinforcement learning) と呼ぶ [30] . XRL 分野の研究としては，入力のうち出力に大きな影響を与える部分を特定するネットワークをシステムの中に組み込む Shi らの研究 [31] や，Huang らによるシステムの挙動を IF-THEN ルールに代替する研究 [32] , 報酬関数  $Q(s, a)$  を複数の副関数に分解する Iucci らの研究 [33] などの様々な試みが存在する .

### 2.2.2 ゲームにおける XAI

McGrath らの研究 [34] ではチェスにおける人間の知識や理論が AlphaZero にどれだけ反映されているかを訓練段階やネットワークの深さなどの多様な指標で調査した . 他にも，Lee ら [35] による AI の着手に対してゲームの固有の知識(ドメイン知識)を用いたモデルの挙動に対する解説文の自動生成の試み等が存在する . しかし，このようなドメイン知識は必ずしも AI の挙動と相関が無いことも指摘されている [34] . AI による画像分類タスクの可視化手法として用いられていた saliency map[26] や Grad-CAM[27] を強化学習に用いる例も存在する [25][36][37] . しかしそれらのニューラルネットワークの活性を根拠とした指標は木探索部分との繋がりが弱く，最終的に決定木を用いて意思決定を行うシステムの動作根拠を直接的に説明できない . また，これらの画像分類用の手法は

- 本来ゲームに存在する時系列の要素を説明に含められない
- 時系列を無視してゲーム画面や盤面の一部を変更する必要がある

という問題点が存在する . また，saliency map や Grad-CAM は主にニューラルネットワークに対する説明手法であり，決定木への応用が困難さも欠点として挙げられる .

### 2.2.3 contrastive explanation

上述の問題点を解決するために，本論文の第 4 章におけるシステムでは AlphaZero が構築する決定木を用いた contrastive explanation(対象説明，比較説

明) を提供する。contrastive explanation は事象を説明する際の方法論の 1 つであり、図 2.5 に示すようにある事象  $a$  が起こった際にその理由を直接説明する代わりに「他の事象  $\bar{a}$  が起らなかった理由」を説明することで間接的にある事象  $a$  の原因を説明するアイデアである [38]。Jacovi らの研究 [38] では自然言語処理の分類タスクにおいて本来の入力データと編集された入力データの出力を比較することで、入力のどの部分がモデルの判断に寄与しているかを示している。Mishra らの研究 [39] では医療タスクにおけるニューラルネットワークの判断を決定木に近似した上で AI の判断  $a$  から派生する予想と別の判断  $\bar{a}$  から派生する予想を提示する形で AI による判断の妥当性を示している。また、Gajcic らの研究 [40] では異なるモデルの挙動の違いをユーザーに示す目的で contrastive explanation が用いられている。これらの強化学習、木構造ネットワークに対する contrastive explanation の手法は決定木上の「最も有力な」「単一の」予想図を用いることを前提としている。そのため第 3 章でしめす提案手法では決定木上の複数の予想図を用いた contrastive explanation の実現を試みた。

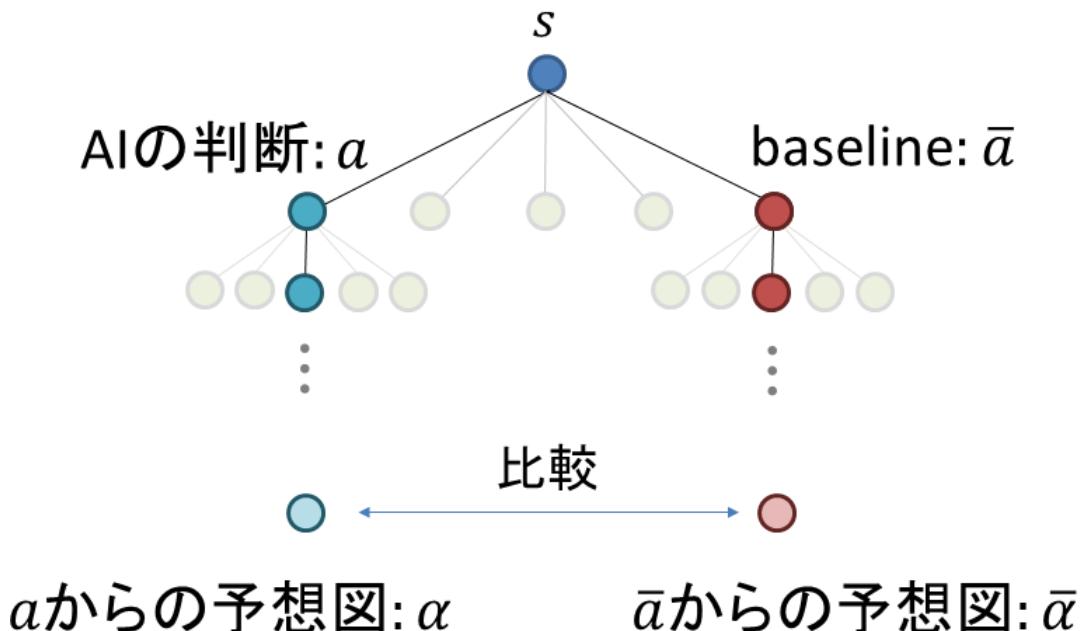


図 2.5: contrastive explanation

### 2.2.4 importance

ゲームやタスクにおいて勝負の分かれ目となりうる場面や，ミスをしやすい場面，危険な事故が起きやすい場面を特定することは非常に有用である。また，このような AI モデルが他の場面よりも重要度が高いと判断する状況を収集することは，モデルの挙動に対する効率的な調査を可能にする。Torray らの研究 [41] や Amir らの研究 [42] では状態の重要度  $I(s)$  を 1 つ先の行動価値関数  $Q(s, a)$  が次の選択によって大きく左右されるような状態を重要度の高い局面として定義している。 $Q(s, a)$  の揺らぎの捉え方にはいくつかの方式が存在し，Torray らは  $I(s)$  を  $s$  から辿り着きうる最善の  $Q(s, a)$  と最悪の  $Q(s, a)$  の差として定義している。

$$I(s) = \max Q(s, a) - \text{worst} Q(s, a) \quad (2.5)$$

Amir らは  $I(s)$  を  $s$  から辿り着きうる最善の  $Q(s, a)$  と 2 番目に値の大きい  $Q(s, a)$  の差として定義している。

$$I(s) = \max Q(s, a) - \text{secondMax} Q(s, a) \quad (2.6)$$

しかし，これらの定義は  $\text{imp}$  が  $Q(s, a)$  の値が低くなるような行動  $a$  に大きく

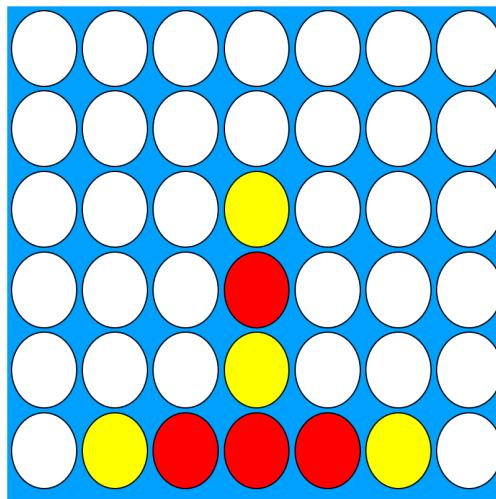


図 2.6: 対称性がある盤面の例

影響されるリスクや、図 2.6 のように状態に対称性があるような場合に重要度が過度に小さくなってしまうリスクがある。

また、Guo らの研究 [43] は重要度  $I$  を収益  $G$  が確定するまでの一連の流れ（エピソード）やエピソード内の時点  $t$  の重要度を収益  $G$  との関連度の大きさから多角的に定めている。しかし、Guo らの定義は状態の符号化や回帰などのデータの加工段階が多く、指標自体の説明性に疑問が残る。AI モデルに対して説明を加える際にはその手法自体もなるべく簡明であり、勝敗に直結している方が望ましいと考えられる。

### 2.2.5 ボードゲーム学習支援

本論文は高度な AI の動作を人間に理解させることを目標としており、学習支援の側面を含んでいると言える。既存の AI を用いたボードゲーム学習支援システムとしては Lee ら [44] やオンラインサービスである DecodeChess[35] などによる解説文自動生成や、Richard[45] らや Richard[46] らの人間側の悪手を自動的に検知しその理由を個別に指摘するモデルが提案されている。しかし「ボードゲームに対する XAI」の段での内容と同様にゲームのドメイン知識に依存しており、指導の内容も人間の知識に依存したものになってしまうという欠点がある。また、指導に特化した人間が挙動に違和感を抱きづらい AI の開発も行われている [47][48]。このアプローチは人間が受け入れやすい方向に AI の側を変更する方式であり、その変更により既存の人間の知識ではとらえられない AI の強みが失われてしまう危険性がある。

## 第3章

### 提案手法の概要

関連研究の章ではニューラルネットワークや決定木が内包する説明性の欠如という問題点と説明を加える既存手法の持つ課題について述べた。そこで、システムの決定木から結果に類似性のある複数の分岐を抽出することによる

- 本来ゲームに存在する時系列の要素を含む
- 評価基準が勝敗に直結する
- 人間のドメイン知識に依存しない

説明手法を提案する。図 3.1 に提案手法の概要を示している。本手法はある状態  $s$  と行動  $a$  の組に対して AI が予想する未来図  $O(s)$  とそこに至るまでの軌跡の集合を取り出し、その傾向を見出すことによる AI の判断の意図を可視化を目標とする。以降は未来図  $O(s)$  と走査が  $O(s)$  に至るまでに通過したノードの履歴(軌跡)  $\zeta(s, O(s))$  を合わせて進行図と記載する。本章では提案手法のアルゴリズムと、第 2 章で述べた重要度  $I(s)$  の新たな定義を記載する。それから、実験のタスクとして選択した connect4 と使用したモデルである alphazero\_baseline、提案手法の connect4 への応用例について記載する。

#### 3.1 提案手法のアルゴリズム

本手法は AlphaZero システムのニューラルネットワークと木探索部分のうち、主に木探索の部分を対象に適用される。本手法では決定木の判断を説明する際

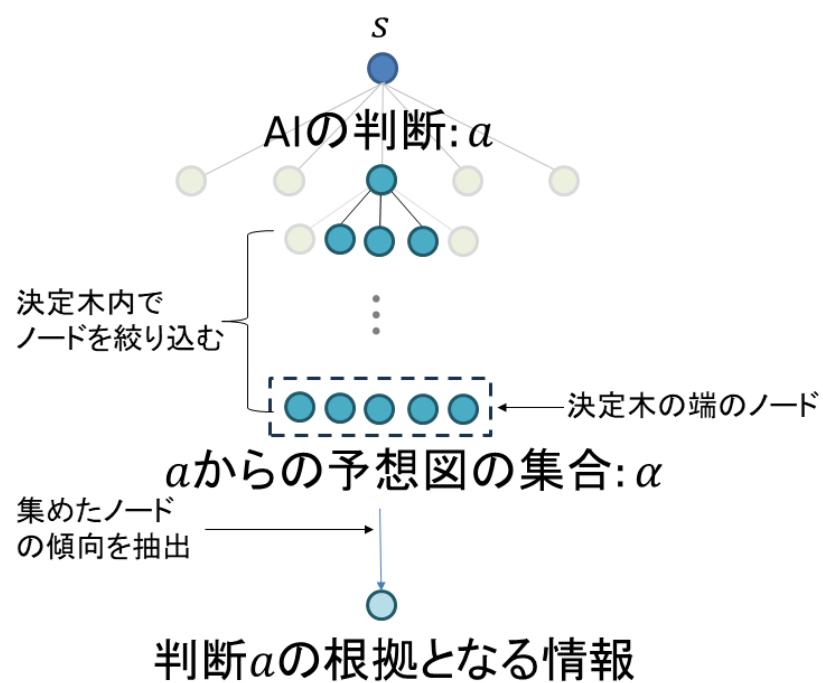


図 3.1: 提案手法の概要

に最も有用な部分を決定木から抽出することを主な目的としており，アルゴリズムはボードゲーム AI に留まらず決定木構造を持つ多くのシステムに応用可能である。アルゴリズムの流れは以下の通りである。図 3.2, 図 3.3 はステップごとのアルゴリズムの概要である。

1. 説明を付与する状態(図 3.2 の青いノード)と行動の組( $s_{start}, a$ )を選択する。このとき注目している状態  $s_{now}, a_{now}$  はそれぞれ

$$s_{now} = s_{start} \quad (3.1)$$

$$a_{now} = a \quad (3.2)$$

とする。

2. 以下の流れを  $l$  ステップ分繰り返す。

決定木中の注目しているノード  $s_{now}$  における訪問回数  $N(s_{now})$  中の上位  $k$  個分の行動  $\{a_1, a_2, \dots, a_k\}$ ( $a_i$  は  $k$  番目に有望な行動とする)を取り出す。 $s_{now}$  から行動  $\{a_1, a_2, \dots, a_k\}$  を取ることでたどり着く各ノード  $\{s_{next_1}, s_{next_2}, \dots, s_{next_k}\}$ ( $s_{next_i} = T(s_{now}, a_i)$ ,  $T$  は遷移関数)に対して同様の操作を繰り返す。図 3.2 に示すように，結果的に  $k^l$  個のノードが取り出される。

3. 集めた  $k$  の  $l$  乗個のノード  $\{s'_1, s'_2, \dots, s'_{k^l}\}$  のそれぞれ  $s'_i (i = 1, 2, \dots, k^l)$  に対して以下の操作を再帰的に繰り返す。 $s'_i$  における方策  $P(s'_i)$  中の最も有望な行動  $a_{promising}$  と  $s'_{next_i} = T(s'_i, a_{promising})$  を記録する。そのようにして記録した  $\{s'_{next_1}, s'_{next_2}, \dots, s'_{next_{k^l}}\}$  のそれぞれ  $s'_{next_j} (j = 1, 2, \dots, k^l)$  にも同様の操作を盤面ノードが決定木の端に辿り着くまで行う。結果的に  $k^l$  個の葉ノード(図 3.3 下部の紫色のノード)が集められる。
4. 3 によってたどり着いた  $k^l$  個のノードによる集合  $S = \{s_{edge_1}, s_{edge_2}, \dots, s_{edge_{k^l}}\}$  を任意の共通項  $c$  によっていくつかの副集合  $\{S_1, S_2, \dots, S_q\}$  に分ける。
5. 共通項で括られた各集合  $\{S_1, S_2, \dots, S_q\}$  のうち，最も要素数が多いもの  $S_{max}$  中の各要素  $\{s_{e_1}, s_{e_2}, \dots, s_{e_u}\}$  と各要素に対応する軌跡  $\{\zeta_{s_{e_1}}, \zeta_{s_{e_2}}, \dots, \zeta_{s_{e_u}}\}$

を抽出する。図 3.3 では青い点線で囲われたグループが抽出される。ここでの軌跡  $\zeta_{s_{e_i}}$  とは、図 3.4 に示すように決定木を走査する際に  $s_{start}$  から  $s_{e_i}$  にたどり着く際に、どのような状態  $s$  や行動  $a$  を経由したかを表している。図 3.4 における  $\zeta_{s_{e_i}}$  は  $\{s_{start}, a_1, s_1, a_2, s_2, a_3, s_3, a_4, s_{e_i}\}$  となる。

① $(s_{start}, a)$ を決定      ②訪問回数上位のノードを収集

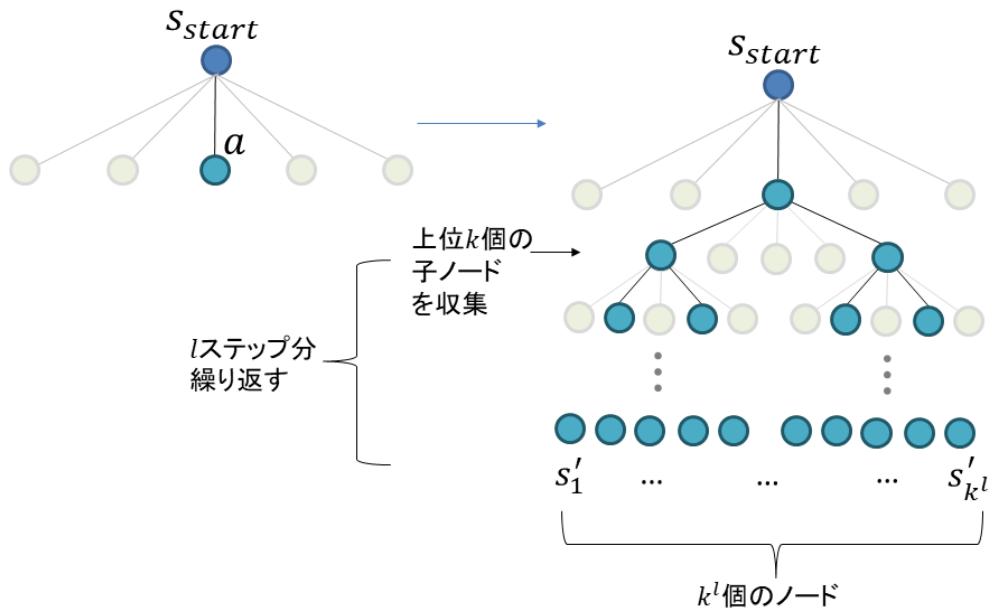


図 3.2: 提案アルゴリズムの概要 (step1~ 2)

このアルゴリズムを要約すると、図 3.5 のようにある局面  $s$  と行動  $a$  の組み合わせから辿り着きやすい結末  $O(s, a)$  を抽出し、同時に  $O(s, a)$  に至るまでの複数の軌跡を抽出すると表現できる。調査を行う者が複数の軌跡を観察することで、共通する傾向や法則性を見出せるというメリットが存在する。これは第 2 章で述べたような最も可能性が高い 1 つの軌跡を示す、といったような情報が単一である従来の手法には不可能である。提案手法の疑似コードは付録 B に記載している。

付録 A で示すように connect4 タスクにおいて収集される軌跡の集合において

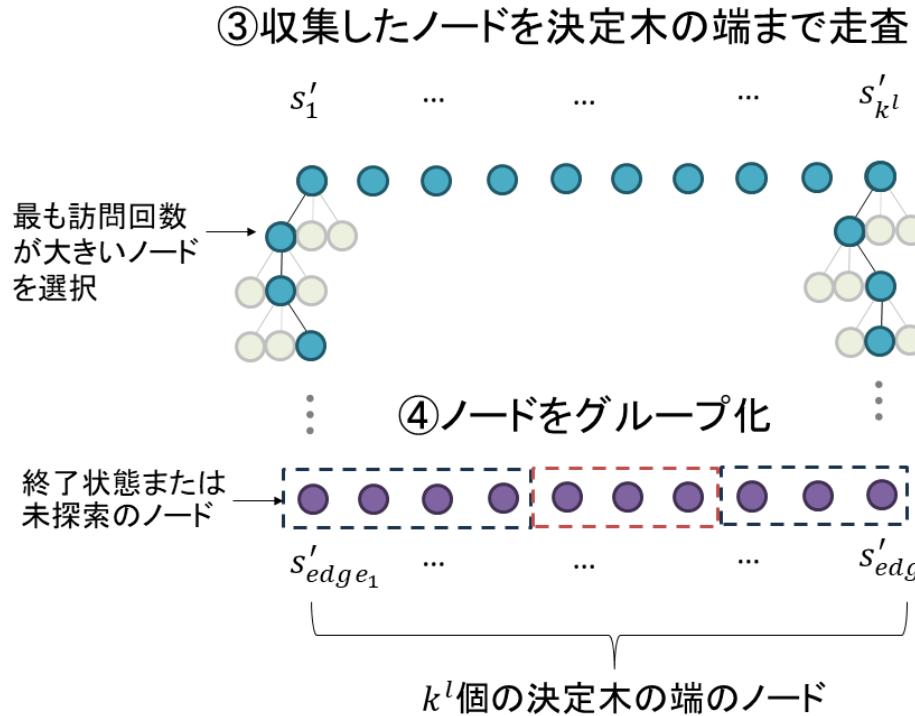


図 3.3: 提案アルゴリズムの概要 (step3~4)

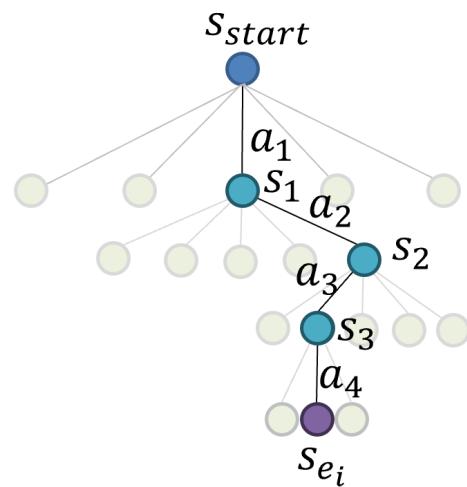


図 3.4: 軌跡の例

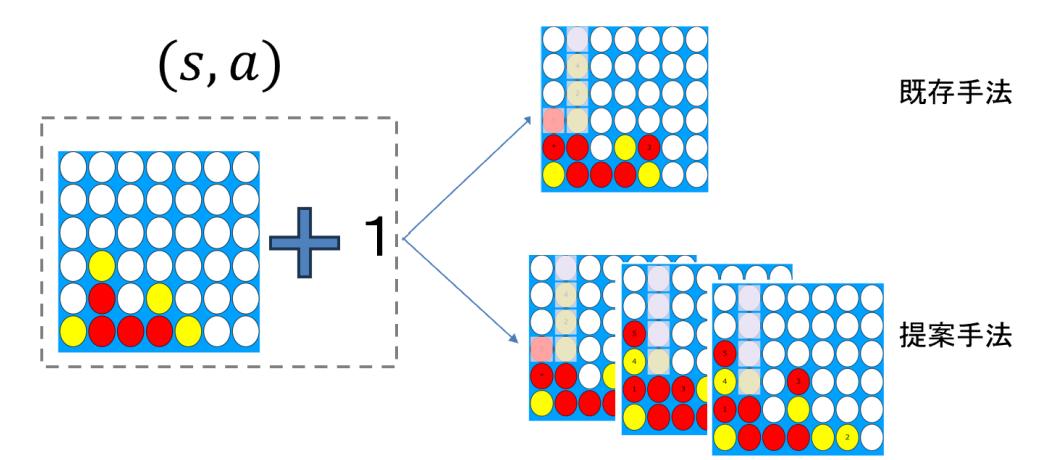


図 3.5: 提案手法と既存手法

は末端部分のいくつかの選択が共通している傾向が確認された。そのため状態と行動の組  $(s, a)$  から  $O(s, a)$  という結果を予測する過程で  $O(s, a)$  から末尾の数手を取り除いた版面  $d$  にたどり着く傾向の発見などの知識獲得が記載される。

### 3.2 importance

前章で述べたように、エピソード中の各状態  $s$  の重要度  $I(s)$  の定義を以下のように定めた。

$$I(s) = V[Q'] \quad (Q' = [\max Q(s, a), \text{secondMax} Q(s, a), \dots, \text{thirdQuantile} Q(s, a)]) \quad (3.3)$$

つまり、現在の状態  $s$  に対する行動の集合  $A (= a_1, a_2, \dots, a_N)$  とした際の行動価値関数の集合  $Q(s, a_1), Q(s, a_2), \dots, Q(s, a_N)$  のうち値の大きさが上位 75 % の成分で構成される集合の分散として重要度  $I(s)$  を定義する。これは次の 1 手で辿り着きうる収益の予想の揺らぎの幅を意味しており、先述した盤面の対称性や悪い選択肢の影響が大きくなる可能性の軽減が期待できる。

### 3.3 connect4

connect4 はボードゲームの一種である。ルールは五目並べに極めて近く、図 3.6 の右側の画像のように、2人のプレイヤーが交互に互いの駒を盤上に置き、最終的に縦、横、もしくは斜めに連続して 4つの石を並べたプレイヤーの勝利となる。ただし、五目並べや連珠との相違点として「重力」の存在が挙げられる。この「重力」とは各プレイヤーが石をボード上の最も下の行または既に置かれた石の上にしか置けないという規則を表している。そのため、各プレイヤーの選択肢はボードの列の数と等しい。図 3.6 の左側は左から 5 番目の列を指定した際に、石がボード上の最下の列に置かれる様子を示している。connect4 の一般的なボードの広さは  $6 \times 7$  であり  $6 \times 7$  のコネクト 4 については 1988 年に Allis[49] により知識ベースの手法による先番勝利が証明された。Tromp[50] による connect4 の  $\alpha - \beta$  木探索によって導かれた各盤面の最善手とそのデータも一般に公開されている。また、connect4 は盤上に全ての情報が開示されており、結果もどちらか一方の勝利または引き分けのみであるため 2 人ゼロ和完全情報ゲーム [51] に分類される。本論文の実験において提案手法を connect4 に適用する際には最終盤面において 4 つ以上並んでいる石の座標を特徴として用いた。

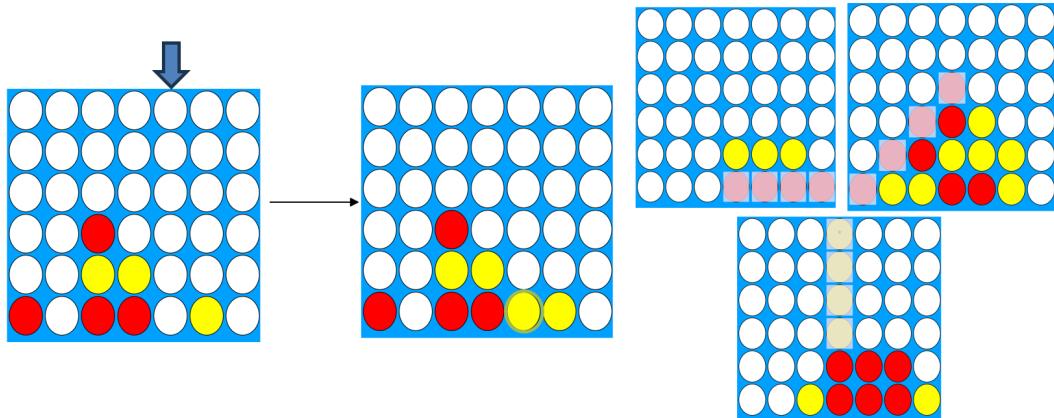


図 3.6: connect4

### 3.4 alphazero\_baseline

alphazero\_baseline は alphaZero を connect4 用に簡易的に模したネットワークであり、図 3.7 に示すように入力は最新の盤面  $s_0$ 、出力は  $1 \times 7$ (7 はボードの列の数) の方策  $P(s)$ (以下  $s = s_0$  とする) とスカラー値の局面評価  $V(s)$ (値域は -1 から 1) である。方策は確率分布であり、要素の値が大きいインデックスを選択することで勝利に近づくと予想される。局面評価は第 2 章の強化学習の段で記載した状態価値関数と同一の変数であり、入力に対する評価を表しており、1 が入力の手番のプレイヤーにとっての勝利、-1 が敗北の予想を表している。

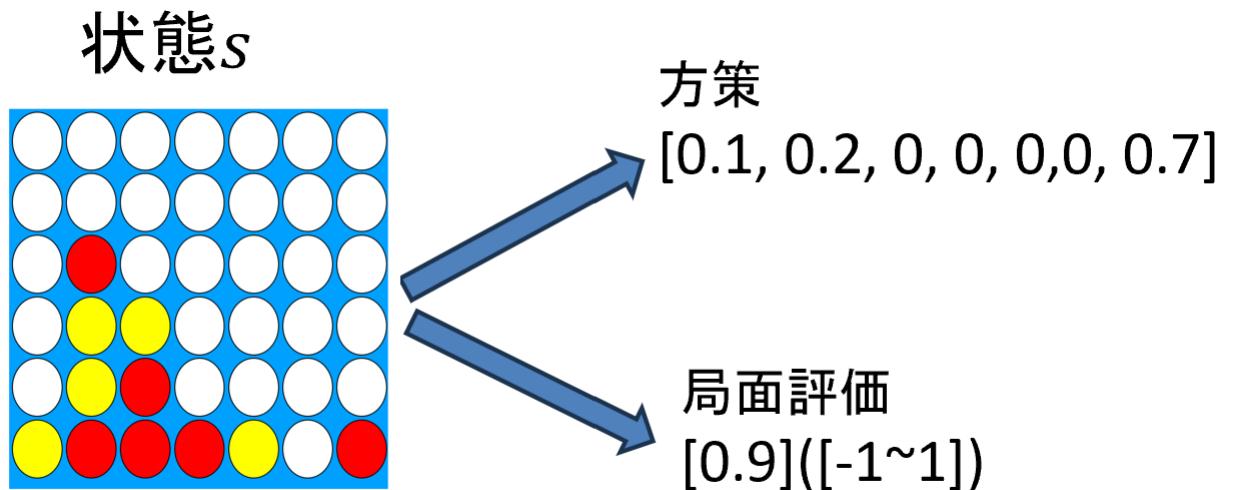


図 3.7: alphazero\_baseline の入出力

### 3.5 提案手法の connect4 への適用

提案手法を connect4 に適用した際の概要を図 3.8 に示す。進行図は 1 組の盤面  $s$  と行動  $a$  に対して決定木中の有力なノードを走査することで生成される。提案手法を connect4 に適用する際、各ノードは alphazero\_baseline への入力である connect4 の各盤面となる。

また、ゲームのルールより収集される最終状態では図 3.8 の下部のように 4 つ以上並んだ石が存在する。第 4 章で詳しく述べるように、実験では 4 つ以上並

んだ石の位置で最終状態をグループ化した。図 3.8においては石が左下から右斜め上方向に 4 つ並んでいる盤面が最も多いため、赤線で囲われた盤面とそこにいたるまでの軌跡が保存される。

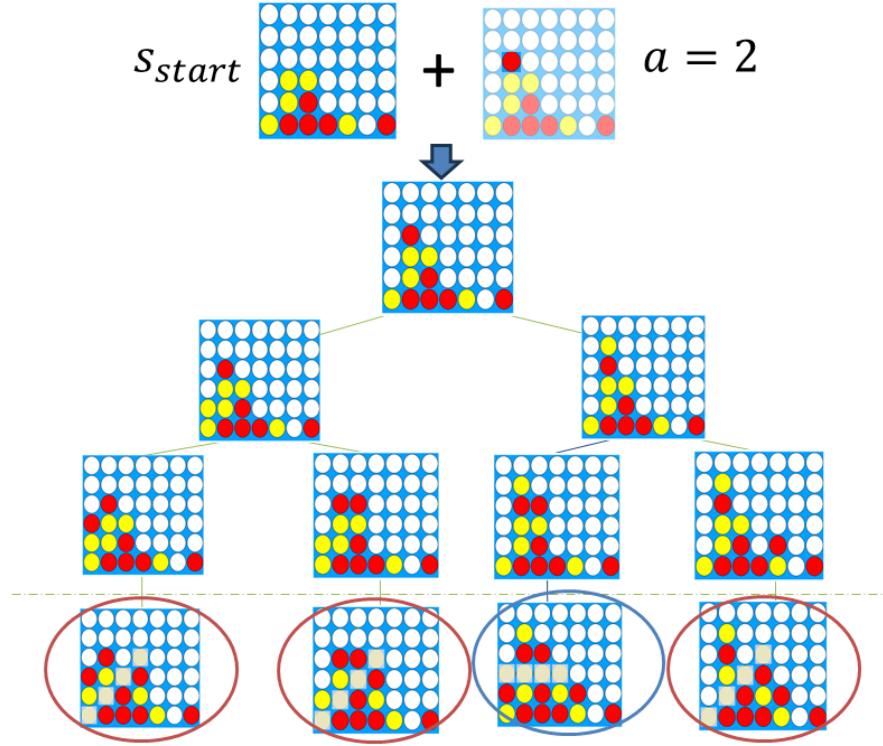


図 3.8: 提案手法の connect4 への適用

提案手法により生成される進行図を図 3.9 に示す。先述のように進行図は予測図と軌跡で構成されている。connect4 では盤面  $s$  と行動  $a$  を選んだ際の次の状態  $s_{next}$  は  $(s, a)$  に対して一意に決定される。そのため、軌跡  $\zeta_{s_{e_i}}$  は  $s_{start}$  から  $s_{e_i}$  に至るまでに取った行動  $a$  の集合とした。各行動は選択した列のインデックスで 1 ~ 7 の数字として表現される。

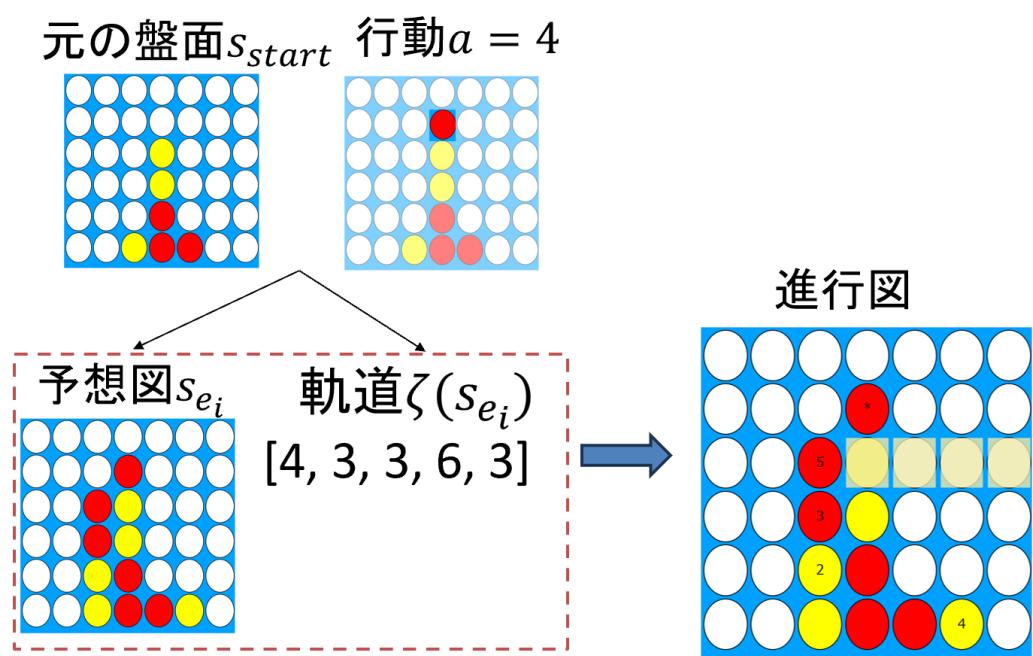


図 3.9: connect4 における進行図

## 第4章

### 評価実験

提案手法の有効性を示すため 2 種類の実験を行った。いずれもタスクの対象として connect4 を扱っている。

1 つ目の実験(以下データ実験と表記)はコンピュータ同士の対戦データを用いて提案手法による想定図の妥当性を評価した。

2 つ目の実験(以下インターフェース実験と表記)は自作の GUI システムを用いた提案手法の学習支援ツールとしての有効性を評価した。本章ではデータ実験、インターフェース実験の詳細と結果を記載する。

#### 4.1 データ実験

提案手法と後に述べる比較手法によるゲーム結果(後で勝敗も含める)の予測精度を比較した。

##### 4.1.1 データセット

alphazero\_baselin モデル同士の対戦データ 2000 局分(盤面数:61049)を使用した。いずれもいずれも弱い AI が先番のデータを使用している。これは弱い AI が後番の場合評価関数の変動が極めて小さくなることと、弱い側が先番を選択することが指導において一般的とされるためである。

#### 4.1.2 データ実験の評価指標

提案手法と次節で述べる比較手法による予測の精度を独自に定義した2つの定義 group prediction rate( $P_g$ ) , stone prediction rate( $P_s$ ) によって計測した . stone prediction rate , group prediction rate の詳細な定義は付録 A に記載する . それぞれが , 最終盤面において 4 つ以上連続して並ぶ石と組み合わせ (以下 fatal stone , fatal group と記載) の座標に対する予測精度を表している .

本論文が提案する提案指標 (group prediction rate , stone prediction rate) の計算において , 盤面の座標は図 4.1 のように定められる . 図 4.2 のようにゲーム

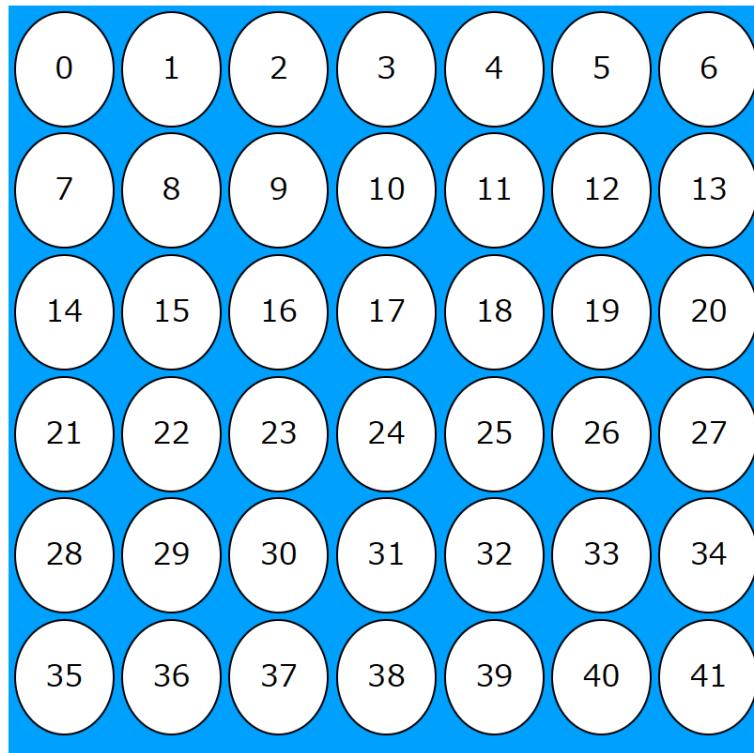
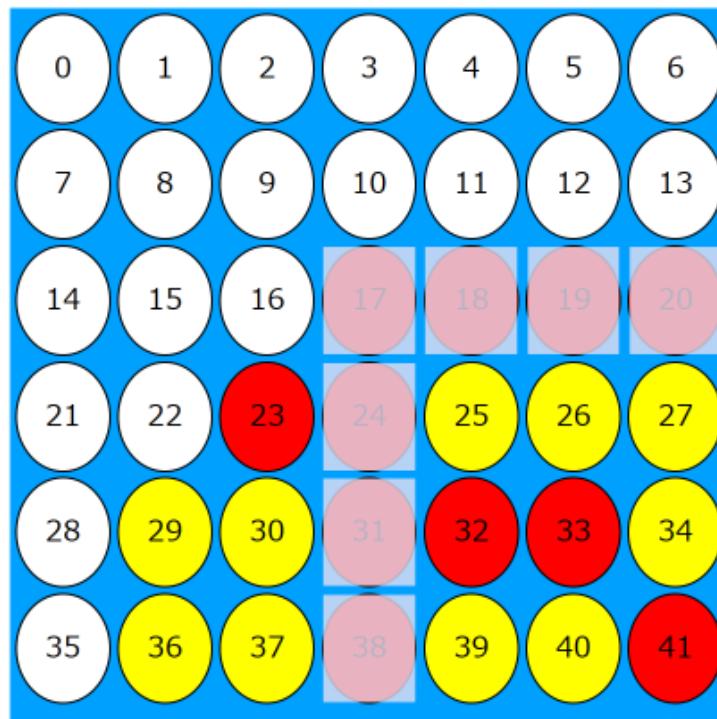


図 4.1: 盤面の座標番号

の終了状態において , 4 つ以上連続してつながっている石の座標  $F_s$ (fatal stone の集合) とその組み合わせ  $F_g$ (fatal group の集合) を記録する . 図 4.2 の盤面は  $\{17, 18, 19, 20, 24, 31, 38\}$  の 7 つの fatal stone と  $\{[17, 18, 19, 20], [17, 24, 31, 38]\}$  の 2 つの fatal group を持つ . group prediction rate( $P_g$ ) , stone prediction rate( $P_s$ )



Fatal Stone: {17, 18, 19, 20, 24, 31, 38}

Fatal Group: {[17, 18, 19, 20], [17, 24, 31, 38]}

図 4.2: fatalStone と fatalGroup

は実際のゲームにおける fatal group と fatal stone の集合 (それぞれ  $R_s, R_g$  とする) と AI の予測による fatal stone と fatal group の集合 ( $F_g, F_s$ ) を比較する指標である。どちらも値が高ければ高い程予測の精度が高いことを表す。

#### 4.1.3 group prediction rate

group prediction rate は fatal group の予測精度を示す 2 値 (0 または 1) の指標である。

$$P_g = 1 \quad \text{If } F_g \cap R_g \neq \emptyset \quad \text{Else} \quad 0 \quad (4.1)$$

例外として  $R_g = \emptyset$  (引き分け) の場合,  $F_g$  も空集合であるならば group prediction rate は 1 となる。

#### 4.1.4 stone prediction rate

stone prediction rate は fatal stone の予測精度を示す。stone prediction rate の値域は  $[0, 1]$  である。(Count() は集合の要素数を数える関数)

$$P_s = \text{Min}\left(\frac{\text{Count}(F_s \cap R_s)}{4}, 1\right) \quad (4.2)$$

例外として  $R_s = \emptyset$  (引き分け) の場合,  $F_s$  も空集合であるならば stone prediction rate は 1 となる。

例えば、実際の結果と予測結果が図 4.3 の様になる場合,  $R_g$  と  $F_g$  は  $[17, 24, 31, 38]$  が共通しているため group prediction rate は 1, stone prediction rate は  $\frac{4}{4} = 1$  となる。

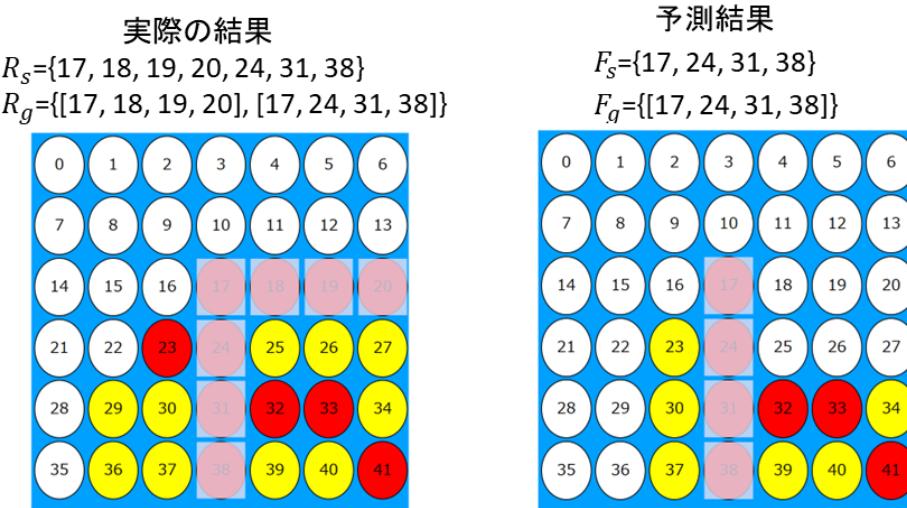


図 4.3: 予測結果の例

#### 4.1.5 比較手法

提案手法と比較手法はそれぞれ以下的方式で予測を行う。比較手法の詳細なアルゴリズムは付録 B に記載した。

- 比較手法: 探索の開始地点から最も訪問回数が大きい選択肢を選び続けたたどり着いた最終局面の  $F_s, F_g$  を用いる .
- 提案手法: 集めた盤面における 4 つ繋がっている石を集計し , 最も出現頻度が高い fatal group を  $F_s$  , 2 番目までに出現頻度が高い fatal group を  $F_g$  として用いる . 組み合わせを 2 つ記録する理由は下の図のように最終的に繋がっている組み合わせが 2 つある可能性を考慮するためである .

図 4.4 は比較手法と提案手法の概要を示している . また , 提案手法による予測

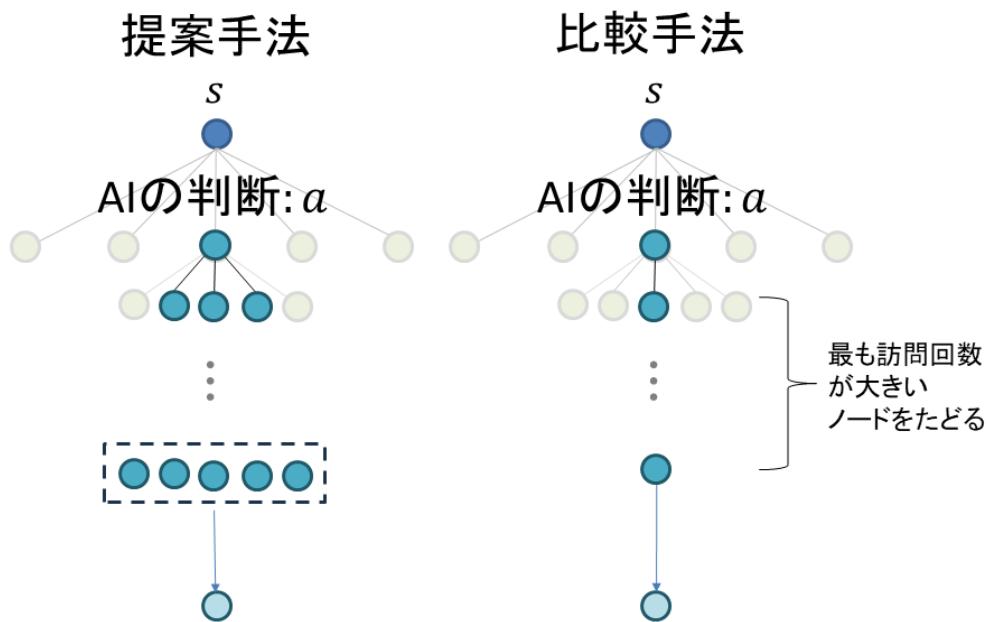


図 4.4: 提案手法と比較手法の比較

が図 4.5 のようになる場合 ,  $[1, 2, 3, 4]$  が最も多く含まれている fatal group であり , 2 番目は  $[1, 9, 17, 25]$  である . そのため , 提案手法全体による予測結果は  $F_g = \{[1, 2, 3, 4], [1, 9, 17, 25]\}$  ,  $F_s = \{1, 9, 17, 25\}$  となる .

分岐1の予測結果	分岐2の予測結果
$F_{g_1} = \{[1, 2, 3, 4], [3, 10, 17, 24]\}$	$F_{g_2} = \{[1, 2, 3, 4]\}$
分岐3の予測結果	分岐4の予測結果
$F_{g_3} = \{[1, 2, 3, 4], [1, 9, 17, 25]\}$	$F_{g_4} = \{[1, 9, 17, 25]\}$

図 4.5: 提案手法による予測

#### 4.1.6 データ実験における提案指標の計測

1. 比較手法: 決定木の走査によりたどり着いた  $F_s, F_g$  によって stone prediction rate , group prediction rate を計算した
2. 提案手法: 第 3 章で述べたアルゴリズムによって収集された最終状態の集合  $S = \{s_{edge_1}, s_{edge_2}, \dots, s_{edge_{k^l}}\}$  を指標ごとにグループ化する.
  - group prediction rate: fatal group によって  $S$  をグループ化してできた集合  $\{S_{g_1}, S_{g_2}, \dots, S_{g_n}\}$  ( $S_{g_i}$  は組み合わせ  $g_i$  が fatal group となっている盤面の集合) を要素が多い順に 2 つ取り出す. 抽出された 2 つの集合  $\{S_{g_{m_1}}, S_{g_{m_2}}\}$  における  $\{g_{m_1}, g_{m_2}\}$  で構成される集合を  $F_g$  とし , group prediction rate を計算した.
  - stone prediction rate:  $g_{m_1}$  を  $F_s$  とし , stone prediction rate を計算した.

#### 4.1.7 実験結果 (データ実験)

盤面データのうち , 手数 (盤面上の赤または黄色) が 13~24 のものに対して実験を行った . 表 4.1 に実験結果 (group prediction rate , stone prediction rate の平均値) を示す . いずれの場合も group prediction rate において提案手法は比較手法より高い値を示した . 表中に記載した「補間」の詳細は付録 B に記載する .

表 4.1: 実験結果:最終状態の予測

	mean group prediction rate	mean stone prediction rate		
手数 (盤面数, 補間の有無)	比較手法	提案手法	比較手法	提案手法
19-24(9862, 無)	0.43	<b>0.60</b>	<b>0.62</b>	0.55
19-24(9862, 有)	0.44	<b>0.63</b>	<b>0.63</b>	0.61
13-24(21022, 無)	0.37	<b>0.52</b>	<b>0.56</b>	0.55
13-24(21022, 有)	0.37	<b>0.55</b>	<b>0.56</b>	0.55

#### 4.1.8 考察 (データ実験)

提案手法は group prediction rate において比較手法より高い結果を示した。この結果は提案手法が fatal group を予測する能力に長けている事を意味する。比較手法によって予測される fatal group の集合は最も訪問回数が大きい軌跡が辿り着く 1 つの最終状態のものである。一方で提案手法により予測される fatal group の集合は、複数の軌跡がそれぞれに辿り着く最終状態の fatal group の多数派投票によって決定され、第 1 位と第 2 位のものが採用される。

使用した対戦データは対戦者間の強さに差があり、予測は強い側の決定木を用いている。決定木内での最も訪問回数の大きい軌跡 (以下主軌跡と記載) は AI が予測する双方が最善を尽くした場合の進行と解釈される。実際に弱い AI と対戦を行う際には、弱い AI は強い AI が最善とみなす行動 (最も訪問回数が大きい行動) を必ずしも選択しない。そのため、実際の進行は、比較手法に用いられる主軌跡とは異なる可能性が高い。提案手法では、主軌跡を含む複数の軌跡を用いているため、より精度の高い予測が可能になったと考えられる。

また、提案手法は 2 組の fatal group を保存することから比較手法よりも予測する fatal group の数が多くなる傾向にあるため、必然的に group prediction rate の値も大きくなったとも考えられる。

## 4.2 インタフェース実験

提案手法の人間に対する有効性を示すため以下のように自作した connect4 の学習支援システムを用いて実験を行った。自作システムの開始画面は 4.6 のように構成されており、右側の青い正方形の部分をクリックすることで connect4 をプレイできる。実験対象者は大学生、大学院生の 22 名（男性 17 名：女性 5 名）となった。

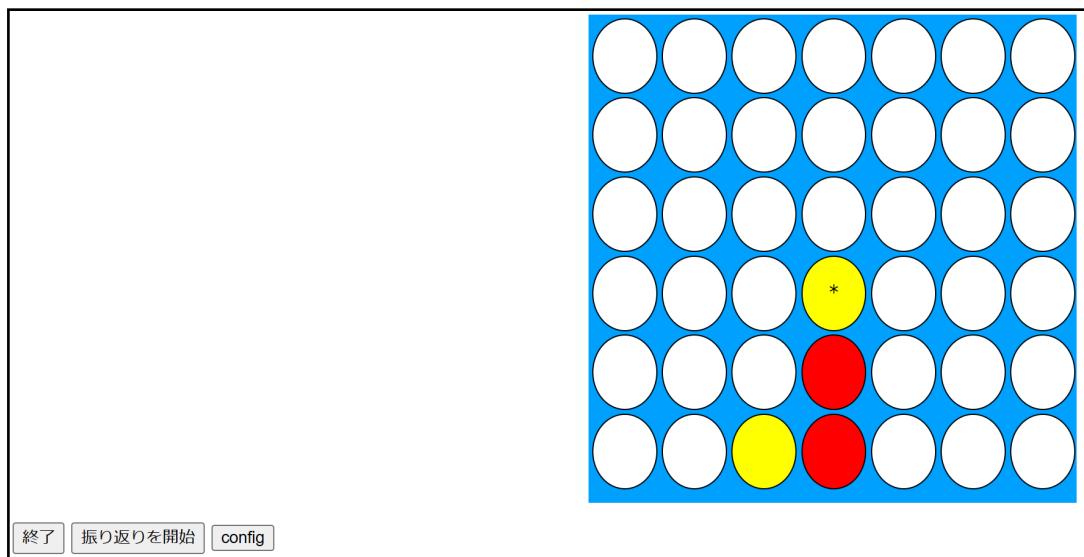


図 4.6: 開始画面

### 4.2.1 実験手順

実験は被験者 1 人あたりにつき 3 回行われ、1 回目と 2 回目を第 1 段階（提案手法による学習）、3 回目を第 2 段階（被験者同士の対戦）とした。ここでは第 1 段階である提案手法の学習について述べる。

#### 4.2.1.0.1 第1段階(提案手法による学習)

提案システムを用いた学習はさらに

- AIシステム(alphazero\_baseline)との対戦
- 提案手法によるゲームの振り返り

の2ステップに分けられる。まず、図4.6の画面をクリックしながらAIとの対戦を行う。実験ではユーザーを先番(赤)とした。

AIシステムとの対戦が終了するとシステムは「振り返りモード」に移行する。「振り返りモード」は図4.7のように構成される。右側に直前のゲームの振り返りたい地点の盤面が表示され、ユーザーは数字の描かれたボタンを押すことでもAIによる進行図を閲覧できる。進行図は提案手法または比較手法によって生成される。インターフェースの詳しい構成は付録Eに記載した。また「振り返りモード」は第3章で定義した重要度  $I(s)$  が最も高い地点から開始する。

実験の際には被験者をグループA(提案手法による進行図を見せるグループ)とグループB(比較手法による進行図を見せるグループ)に分類した。数字が描かれたボタンは列のインデックスを表しており、図4.8に示すように各ボタンを押した際にその列を選択した場合の想定図と局面評価を確認できる。図4.8では「4」と描かれたボタンを押したため4列目を選択した場合の進行図が表示されている。

#### 4.2.2 インタフェース実験の評価指標

インターフェース実験の評価指標は主観評価と客観評価の2つに分けられる。主観評価は被験者による五段階評価であり、「タスクの熟達度に関する質問」((a))と「タスクの楽しさや面白さに関する質問」((b))の2つに分けられる。具体的な質問事項は付録Dに記載する。客観評価はグループA(提案手法)の被験者とグループB(比較手法)の被験者の対戦成績である。

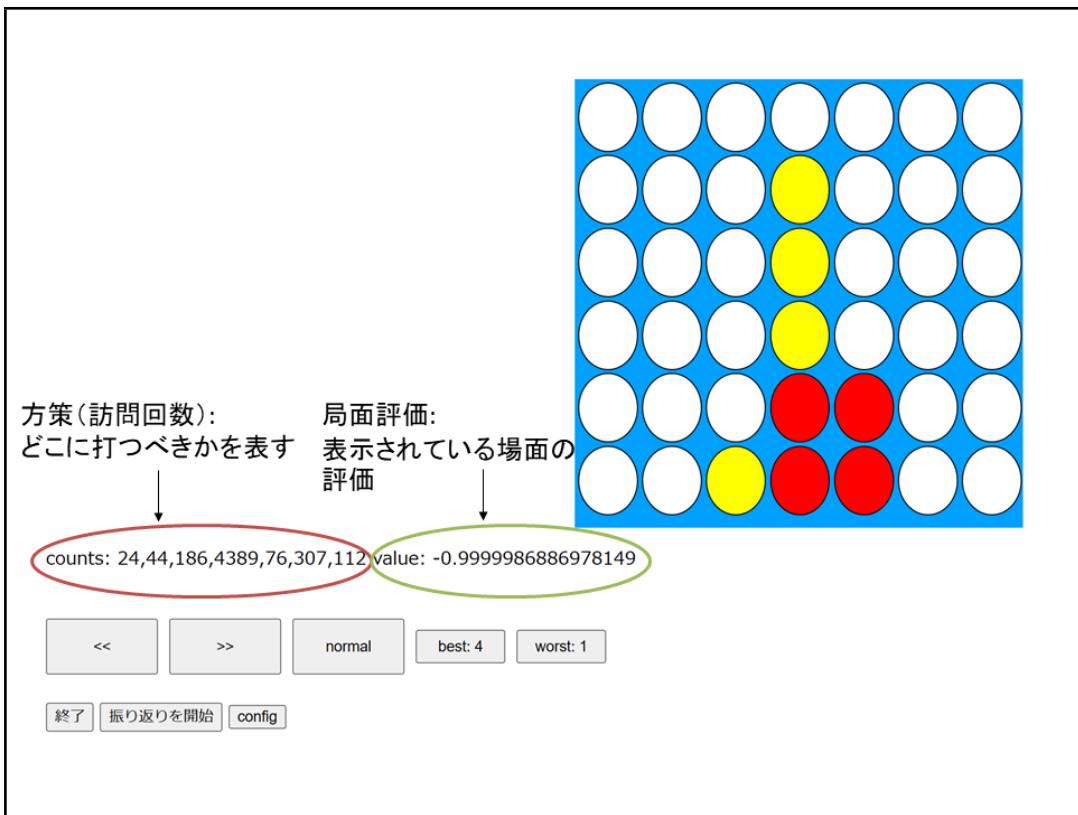


図 4.7: 振り返り画面

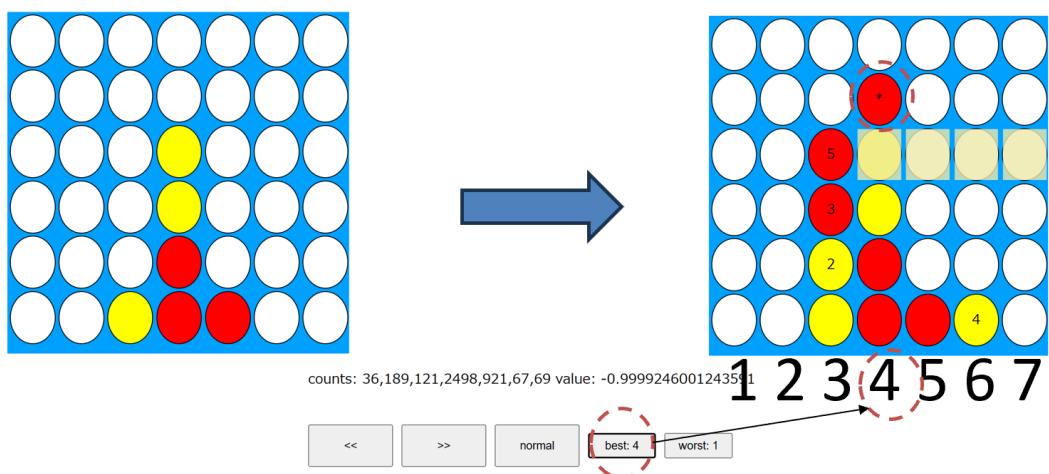


図 4.8: 進行図の表示

### 4.2.3 実験結果(インターフェース実験)

進行図で提示される先読みの手数ごとに主観評価の平均値(1~5)を表4.2, 表4.3, 表4.4, 表4.5に示す。先読みの手数は3, 5, 7, 制限なしの4種類であり、先読みの手数ごとに図4.9に示すように提示される進行図中の手数が変化する。先読みの手数に関わらず、AIが予測する進行において最終的に4つ以上の石が並ぶ座標は強調表示される。先読み手数が3手または先読み手数制限なしの場合

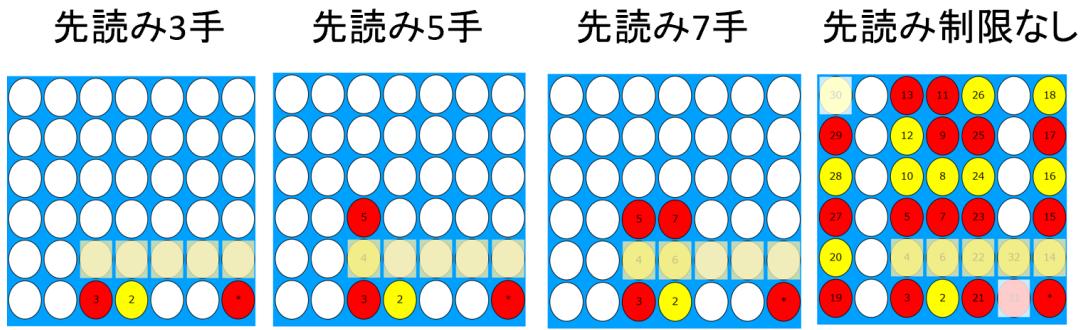


図4.9: 進行図の表示

合に、提案手法は比較手法に対して高い結果を示した。表4.2, 表4.3, 表4.4, 表4.5に対してt検定を行ったが、いずれも有意な差は見当たらなかった。

表4.2: 先読み手数3手の場合(インターフェース実験)

質問		主観評価	
質問の種類	質問の内容	B(比較手法)	A(提案手法)
(a)	振り返りによって成長したと感じましたか	3.17	<b>3.29</b>
	振り返りによってAIの意図を掴むことができたと感じますか	<b>2.67</b>	2.43
	(2回目のみ)1回目に比べて成長したと感じますか	<b>4.33</b>	2.75
(b)	今後もconnect4をプレイするときにこのシステムを使いたいと思いましたか	3.67	<b>3.71</b>
	システムにより振り返りが楽しくなったと感じますか	<b>3.60</b>	3.57

表4.3: 先読み手数5手の場合(インターフェース実験)

質問		主観評価	
質問の種類	質問の内容	B(比較手法)	A(提案手法)
(a)	振り返りによって成長したと感じましたか	4.00	3.00
	振り返りによってAIの意図を掴むことができたと感じますか	4.00	2.60
	(2回目のみ)1回目に比べて成長したと感じますか	<b>4.33</b>	3.67
(b)	今後もconnect4をプレイするときにこのシステムを使いたいと思いましたか	<b>4.80</b>	4.00
	システムにより振り返りが楽しくなったと感じますか	<b>4.60</b>	4.20

表 4.4: 先読み手数 7 手の場合 (インターフェース実験)

質問		主観評価	
質問の種類	質問の内容	B(比較手法)	A(提案手法)
(a)	振り返りによって成長したと感じましたか	4.33	3.83
	振り返りによって AI の意図を掴むことができたと感じますか	3.67	2.67
	(2 回目のみ) 1 回目に比べて成長したと感じますか	4.33	2.75
(b)	今後も connect4 をプレイするときにこのシステムを使いたいと思いましたか	4.67	4.17
	システムにより振り返りが楽しくなったと感じますか	3.50	3.00

表 4.5: 先読み手数制限なしの場合 (インターフェース実験)

質問		主観評価	
質問の種類	質問の内容	B(比較手法)	A(提案手法)
(a)	振り返りによって成長したと感じましたか	4.00	4.25
	振り返りによって AI の意図を掴むことができたと感じますか	3.00	3.50
	(2 回目のみ) 1 回目に比べて成長したと感じますか	4.33	2.75
(b)	今後も connect4 をプレイするときにこのシステムを使いたいと思いましたか	4.00	4.5
	システムにより振り返りが楽しくなったと感じますか	4.40	4.75

また，第 2 段階である被験者同士の対戦結果を表 4.6 に示す．付録 D に示すように被験者 1 人に対して 2 回対戦を行った．結果として示す score は勝利を 1，敗北を -1，引き分けを 0 とした 2 回の対戦の平均値として定義される．提案手法を用いて訓練を行ったグループ (A) は比較手法を用いて訓練を行ったグループ (B) よりも高い勝率を記録した．

表 4.6: 対戦の結果 (インターフェース実験)

グループ	B(比較手法)	A(提案手法)
mean score	-0.15	0.15

また，第 3 章で示した新たな重要度の定義 (式 3.3) の妥当性を検証するため，盤面  $s$  の重要度  $I(s)$  と被験者が  $s$  の進行図を見た時間  $t$  の相関を調査した．比較手法には式 2.6 の定義を用いた．表 4.7 に示したように，いずれの手法による重要度も  $t$  との有意な相関が見られなかった．

表 4.7: 手法別の重要度と  $t$  の相関 (インターフェース実験)

日数	比較手法	提案手法
1 日目	-0.034	-0.035
2 日目	0.067	-0.083

#### 4.2.4 考察 (インターフェース実験)

提案手法を用いたグループ A がグループ B を上回る評価を得たのは、主に先読みが最も短い 3 手の場合と最も長い制限なしの場合となった。

先読み 3 手、制限なしの場合のデータをさらに 1 日目、2 日目で分類した結果、

- 先読み 3 手の場合は 1 日目
- 先読み制限なしの場合は 2 日目

の方がよりグループ A からの評価が高いことが分かった。付録 E で述べるように、1 日目と 2 日目では GUI の設定に差があり 1 日目の GUI では全ての選択肢 (最大 7 種) を起点とした進行図を閲覧可能であるのに対し、2 日目の GUI では起点となる選択肢を 2 つに制限した。そのため、2 日目は被験者が見られる進行図の数が 1 日目の約  $\frac{1}{3}$  となる。

そのため、「1 日目に先読み 3 手の場合」と「2 日目に先読み制限なし」の場合に被験者が受け取る情報量は近しいと推測される。

また、データを「被験者のグループ、日にち、先読み手数」によって分類し、「振り返りによって AI の意図を掴むことができたと感じますか」という質問に対する評価 (以下把握満足度と表記) の降順に並べ直した結果を表 4.8 に示す。

表 4.8: 把握満足度 (インターフェース実験)

グループ	日	先読み手数	把握満足度
B	2	5	4.67
B	2	7	4.00
A	1	制限なし	3.50
A	2	制限なし	3.50
B	1	3	3.50
B	1	制限なし	3.33
A	1	3	3.33
( ... )			
A	2	3	1.75
B	1	3	1.50

グループ A のデータを日にちと先読み手数で分けた場合最も把握満足度が高いのは「2 日目に先読み制限なし」，次いで「1 日目に先読み制限なし」，「1 日目に先読み 3 手」となった。最も情報が少ない「2 日目に 3 手」はグループ A の全 8 パターン中最下位となった。

グループ B に対しても同様に「振り返りによって AI の意図を掴むことができたと感じますか」への評価を集計した結果，最も評価が高いのは「2 日目に 5 手」，次いで「2 日目に 7 手」，「1 日目に 3 手」となった。

また，付録 A の表 A.3 より提案手法が示す 1 組の状態  $s$ ，行動  $a$  について示す軌跡の数  $n$  の平均は約 10 であるため，比較手法が示す情報量は提案手法の約  $\frac{1}{10}$  となる。また，先読みの手数に制限が無い場合の手数の平均は約 15 手である。そのため提示する情報量が大きい程評価が高くなるわけではなく，手法ごとに被験者が適切と感じる情報量が存在すると推測される。

以上の考察を踏まえて，被験者に対して提示する情報を最小化した上で適切な情報量を調査するため，先読みを 1~5 手として再実験を行った。インターフェース実験では補助として提示した訪問回数，局面評価に注目する被験者が多く見られたため，訪問回数「count」(訪問回数) の表示を廃止し「value」(局面評価) と進行図の閲覧は被験者の手番(先番)となる盤面でのみ可能とした。

結果を 4.9 に示す。

表 4.9: 先読み手数制限なしの場合 (追加実験)

質問の種類	質問の内容	主観評価	
		B(比較手法)	A (提案手法)
(a)	振り返りによって成長したと感じましたか	<b>3.80</b>	3.00
	振り返りによって AI の意図を掴むことができたと感じますか	2.80	<b>3.00</b>
	振り返りでは対戦を振り返るのに十分な量の情報を受け取ることができたと思いますか	3.00	<b>3.60</b>
(b)	今後も connect4 をプレイするときにこのシステムを使いたいと思いましたか	3.80	<b>4.2</b>
	システムにより振り返りが楽しくなったと感じますか	<b>4.20</b>	3.6

把握満足度と「充分な情報量」の平均は提案手法の方が高くなつた。そのため、被験者が比較手法では情報量が不十分であると感じている際に、提案手法で情報を補足することには一定の意義があると推測される。ただし、t検定による有意差は見当たらなかつた。また、先読みが1手や2手の場合は、提案手法による軌跡が枝分かれせず、結果的に表示される軌跡の数が小さくなる。そのため、提案手法と比較手法の差異は「決定木中のどの部分を取り出すか」の差異となる。付録BのB.5で述べたように提案手法で示す進行図は盤面の局面評価と符合するように調整されていることも評価に寄与したと考えられる。

# 第5章

## 結論

本論文では

- 決定木から有力なノードを抽出
- 収集されたノードのさらなるグループ化
- 決定木中の有力なノードへの軌跡の保存と提示

により、強化学習 AI システムの判断の可視化を試みた。提案手法の有効性を示すため、connect4(ボードゲーム)を題材として 2 つの実験を行った。第 1 の実験では AI 同士の対戦データを用いた提案手法の予測の妥当性を評価し、実際に比較手法よりも指標によっては優れた予測精度を示した。第 2 の実験では自作の GUI システムを用いた提案手法の有用性を検証し、ユーザーに提示される情報量が極端に大きい場合または小さい場合に比較手法よりも高いユーザー評価を得うることがわかった。

課題としては、第 4 章において提案手法を connect4 に適用する際に「石を 4 つ並べたプレイヤーが勝利する」というゲームのルールに基づき「4 つ並んだ石の座標」を用いてノードのグループ化を行っていることが挙げられる。この事はアルゴリズムの一部でドメイン知識を用いている事を意味し、さらなる汎用化の余地が存在する。また、インターフェースとしてもユーザーに提示する情報の量や種類もさらなる調査の必要性を感じた。そのため、今後は

- 手法のゲーム固有の知識(ドメイン知識)からの脱却
- ユーザーの学習支援に適切な情報提示の方式の模索

## 結論

---

に取り組みたい。

## 謝辞

本研究を行うにあたり親身に相談に乗っていただき，ご指導してくださった  
萩原将文教授，ならびに共に問題解決，議論，相談，および実験に付き合って  
くださった研究室の先輩方，同期の皆様，実験に参加してくださった大学の友人  
達に深く感謝いたします．誠にありがとうございました．

## 参考文献

- [1] *AIに敗れた李九段、「アルファ碁」に教わったこと*. 2016. URL: <https://www.nikkei.com/article/DGXMZ009064900S6A101C1I00000/>.
- [2] “5月11日米IBMのスパコン、チェス世界王者に勝利”. In: 日本経済新聞 (2019). URL: <https://www.nikkei.com/article/DGKKZ044613220Q9A510C1EAC000/>.
- [3] “電王戦、今年で終了「歴史的な役割終えた」”. In: 每日新聞 (2017). URL: <https://mainichi.jp/articles/20170223/k00/00m/040/043000c>.
- [4] et al. SILVER David. “Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm”. In: *CoRR* abs/1712.01815 (2017).
- [5] James Dow Allen. *The Complete Book of Connect 4*. Puzzle Wright Press, 2010.
- [6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60 (2012), pp. 84 –90.
- [7] Robin Rombach et al. “High-Resolution Image Synthesis with Latent Diffusion Models”. In: *CoRR* abs/2112.10752 (2021).
- [8] Long Ouyang et al. *Training language models to follow instructions with human feedback*. 2022. arXiv: 2203.02155 [cs.CL].
- [9] *G7 Hiroshima Leaders' Communiqué*. 2023. URL: <https://www.g7hiroshima.go.jp/en/documents/>.

- [10] 厚生労働省. 新しい資本主義のグランドデザイン及び実行計画 2023 改訂版. 2023. URL: [https://www.cas.go.jp/seisaku/atarashii\\_sihonsyugi/pdf/ap2023.pdf](https://www.cas.go.jp/seisaku/atarashii_sihonsyugi/pdf/ap2023.pdf).
- [11] パンダネット. 囲碁 AI 検討サービス「AIの一歩」. URL: [https://www.pandanet.co.jp/study/ai\\_no\\_itte/](https://www.pandanet.co.jp/study/ai_no_itte/).
- [12] 将棋ウォーズ. 棋神. URL: <https://shogiwars.heroz.jp/guides/beginners/kishin?locale=ja>.
- [13] Richard S. Sutton and Andrew G. Barto. *reinforcement learning*. The MIT Press, 2018, pp. 251–272.
- [14] Volodymyr Mnih et al. “Playing atari with deep reinforcement learning”. In: *arXiv preprint arXiv:1312.5602* (2013).
- [15] et al. Hessel Matteo. “Rainbow: Combining improvements in deep reinforcement learning”. In: *Proceedings of the AAAI conference on artificial intelligence* 32.1 (2018).
- [16] et al. SILVER David. “Mastering the game of Go with deep neural networks and tree search”. In: *nature* 529.7587 (2016).
- [17] Stockfish 12. 2020. URL: <https://stockfishchess.org/blog/2020/stockfish-12/>.
- [18] T. Yamaoka. *DeepLearningShogi*. URL: <https://github.com/TadaoYamaoka/DeepLearningShogi>.
- [19] Bo Zhou. *AlphaZero baseline - ConnectX*. 2020. URL: <https://github.com/PaddlePaddle/PARL/tree/develop/benchmark/torch/AlphaZero>.
- [20] Stockfish 12. 2023. URL: <https://stockfishchess.org/>.
- [21] コンピュータ将棋ソフト「elmo」導入方法. 2020. URL: [https://mk-takizawa.github.io/elmo/howtouse\\_elmo.html](https://mk-takizawa.github.io/elmo/howtouse_elmo.html).
- [22] Stockfish 13. 2021. URL: <https://stockfishchess.org/blog/2021/stockfish-13/>.

- [23] Heleen Rutjes, Martijn C. Willemse, and Wijnand A. IJsselsteijn. “Considerations on Explainable AI and Users’ Mental Models”. In: *CHI conference* (2019).
- [24] Finale Doshi-Velez and Been Kim. “Towards a rigorous science of interpretable machine learning”. In: *arXiv preprint arXiv:1702.08608* (2017).
- [25] Tobias Huber et al. “Local and Global Explanations of Agent Behavior: Integrating Strategy Summaries with Saliency Maps”. In: *Artificial Intelligence* 301.103571 (2021).
- [26] Hou, Xiaodi, and Liqing Zhang. “Saliency detection: A spectral residual approach”. In: *2007 IEEE Conference on computer vision and pattern recognition* (2007).
- [27] et al Selvaraju Ramprasaath R. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”. In: *Proceedings of the IEEE international conference on computer vision* (2017).
- [28] Pang Wei W Koh et al. “On the accuracy of influence functions for measuring group effects”. In: *Advances in neural information processing systems* 32 (2019).
- [29] Marco Tulio Correia Ribeiro. *lime*. 2016. URL: <https://github.com/marcotcr/lime>.
- [30] Yanzhe Bekkemoen. “Explainable reinforcement learning (XRL): a systematic literature review and taxonomy”. In: *Machine Learning* (2023), pp. 1–87.
- [31] Wenjie Shi et al. “Self-Supervised Discovering of Causal Features: Towards Interpretable Reinforcement Learning”. In: *CoRR* abs/2003.07069 (2020).
- [32] Jianfeng Huang, Plamen P. Angelov, and Chengliang Yin. “Interpretable policies for reinforcement learning by empirical fuzzy sets”. In: *Eng. Appl.*

- Artif. Intell.* 91 (2020), p. 103559. URL: <https://api.semanticscholar.org/CorpusID:212766105>.
- [33] Alessandro Iucci et al. “Explainable Reinforcement Learning for Human-Robot Collaboration”. In: (2021), pp. 927–934.
- [34] et al. McGrath Thomas. “Acquisition of Chess Knowledge in AlphaZero”. In: *Proceedings of the National Academy of Sciences* 119.47 119.47 (2022).
- [35] et al. Zeev Fine Ofer Shamai. *DecodeChess*. 2017. URL: <https://decodechess.com/>.
- [36] Anurag Koul Sam Greydanus and Alan Fern Jonathan Dodge. “Visualizing and Understanding Atari Agents”. In: *International conference on machine learning* (2018).
- [37] Yuanfeng Pang and Takeshi Ito. “Visualizing and Understanding Policy Networks of Computer Go”. In: *Journal of Information Processing* 29 (2021).
- [38] Alon Jacovi et al. “Contrastive explanations for model interpretability”. In: *arXiv preprint arXiv:2103.01378* (2021).
- [39] Aditi Mishra et al. “Why? why not? when? visual explanations of agent behaviour in reinforcement learning”. In: (2022), pp. 111–120.
- [40] Jasmina Gajcin et al. “Contrastive explanations for comparing preferences of reinforcement learning agents”. In: *arXiv preprint arXiv:2112.09462* (2021).
- [41] Lisa Torrey and Matthew Taylor. “Teaching on a budget: Agents advising agents in reinforcement learning”. In: *In Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems* (2013).
- [42] Ece Kamar Ofra Amir and Barbara J. Grosz Andrey Kolobov. “Interactive teaching strategies for agent training”. In: *International Joint Conferences on Artificial Intelligence* (2016).

- [43] Xian Wu Wenbo Guo and Xinyu Xing Usmann Khan. “EDGE: Explaining Deep Reinforcement Learning Policies”. In: *Advances in Neural Information Processing Systems 34* (2021).
- [44] Andrew Lee et al. “Improving Chess Commentaries by Combining Language Models with Symbolic Reasoning Engines”. In: *arXiv preprint arXiv:2212.08195* (2022).
- [45] Simon Viennot1 Kokolo Ikeda1 and Naoyuki Sato1. “Detection and labeling of bad moves for coaching go”. In: *2016 IEEE Conference on Computational Intelligence and Games (CIG)* (2016).
- [46] Masahiko Osawa Vincent Richard and Michita Imai. “Determining Strategies behind Moves in the Game of Go”. In: *Cloud Network Robotics 117.95* (2017).
- [47] 伊藤毅志 仲道隆史. “プレイヤの技能に動的に合わせるシステムの提案と評価”. In: *情報処理学会論文誌* 57.11 (2016).
- [48] Reid McIlroy-Young et al. “Aligning superhuman ai with human behavior: Chess as a model system”. In: (2020), pp. 1677–1687.
- [49] L.V. Allis. *A Knowledge-based Approach of Connect-four: The Game is Solved: White Wins*. Technical reports in computer science. University of Limburg, Department of Computer Science, 1992. URL: <https://books.google.co.jp/books?id=1iz0GgAACAAJ>.
- [50] John Tromp. *Connect-4 Data Set*. 1995. URL: <https://archive.ics.uci.edu/dataset/26/connect+4>.
- [51] 毅志 伊藤, 邦仁 保木, and 陽一郎 三宅. ゲーム情報学概論 : ゲームを切り拓く人工知能. コロナ社, 2018.
- [52] Peter Cnudde. *1k connect4 validation set*. 2020. URL: <https://www.kaggle.com/petercnudde>.

## 参考文献

---

- [53] Peter Cnudde. *Scoring connect-x agents*. 2020. URL: <https://www.kaggle.com/code/petercnudde/scoring-connect-x-agents/notebook>.

## 付録 A

### データ実験の詳細

#### A.1 使用したモデルの詳細

第3章で述べたとおり使用した対戦データは弱いAIを先番とし、強いAIを後手としている。AIの強さは一手ごとの探索を行う時間(time), 付録Cで述べる $C_{puct}$ , ニューラルネットワークの訓練段階におけるエポック数の値によって調整した。timeと $C_{puct}$ , エポック数はいずれも値が大きい程モデルは強くなると考えられる。(エポック数については付録Cを参照) 対戦データ生成時のパラメータは表A.1の幅からゲームごとにランダムな値を採用した。これはパラメータの値を変化させることでゲームデータに多様性を持たせるためである。

表 A.1: 対戦データのパラメータ

モデル	強	弱
time	3-5	0-2
$C_{puct}$	0.8-1	0-0.5
エポック数	200	1

#### A.2 対戦結果の詳細

2000ゲームのうち1983ゲームは強いAI(後番)の勝利となった。また、ゲームごとの手数は75%のゲームが36手以内で終了している。そのため、データ実

験ではゲームの中盤と言える 13~24 手目のデータを使用した。図 A.1 にゲームの終了手数の累計グラフを示す。

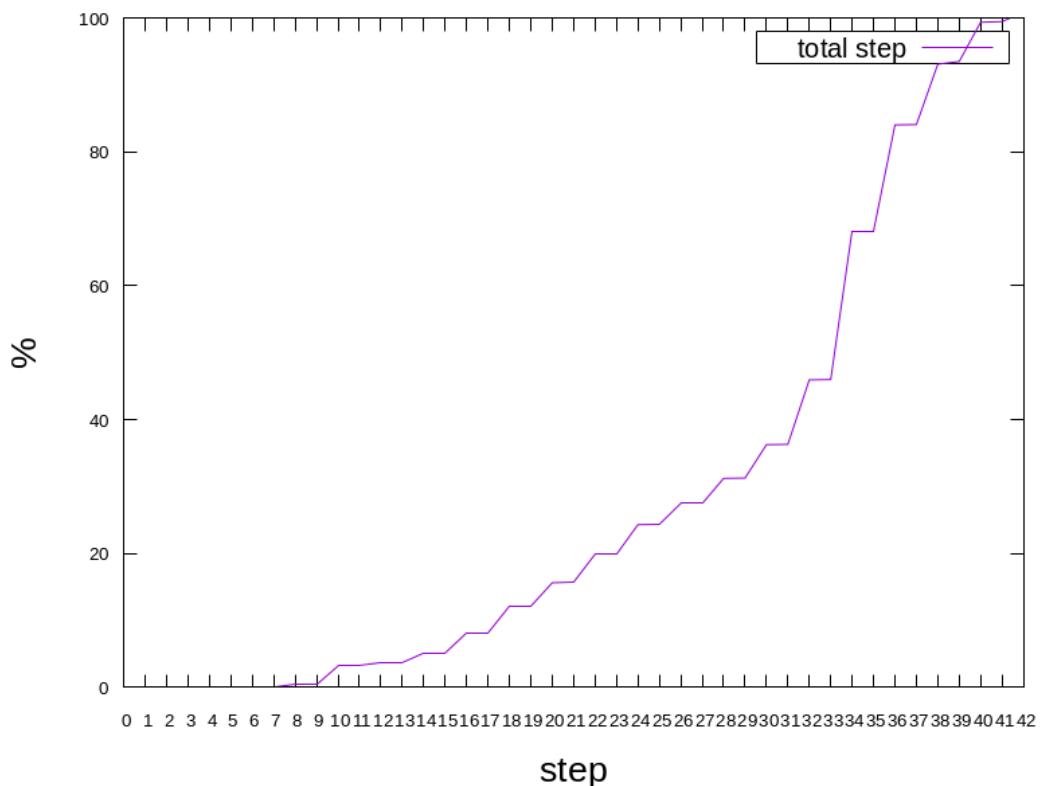


図 A.1: 終了手数の分布

### A.3 データ実験に使用したモデルの詳細

第 4 章に記載した表 4.1 の結果を求める際に用いたモデルのパラメータを表 A.2 に示す。

表 A.2: データ実験: 使用モデルのパラメータ

パラメータ名	値
time	対戦時の後番のモデルと同じ
$C_{puct}$	対戦時の後番のモデルと同じ
エポック数	200
$k$ (提案手法のみ)	4
$l$ (提案手法のみ)	2

表 A.2 の通り、対戦データが生成された際のモデルのパラメータを使用している。そのため本手法はモデルの構造とパラメータの値へのアクセスが可能な場合を想定したホワイトボックス的アプローチの実験であると言える。モデルの time,  $C_{puct}$  の値を固定して同様の手法を適用した場合の実験結果を A.5 で示す。

## A.4 末尾のグループ化

提案手法は予想図とその予想図に至る軌跡(両方を合わせて進行図と記載)を抽出する。図 A.2 は軌跡を末尾の 3 手で 2 つのグループに再編する様子を表している。(ここでは軌跡は行動  $a$  の連続として表している) 取り出された軌跡を

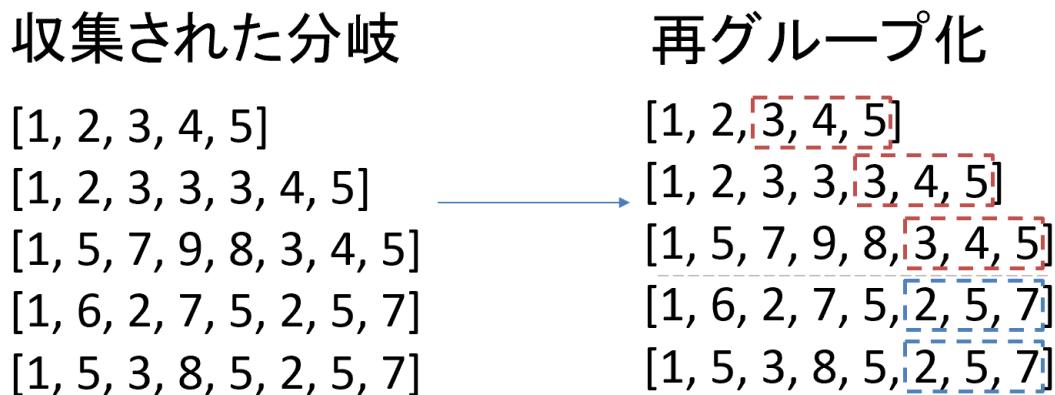


図 A.2: 再グループ化

末尾の 3 手の選択でさらにグループ化した結果、元の軌跡の数と再グループ化

された軌跡の数の平均は表 A.3 のようになった。結果として、取り出された軌跡のうち、平均約 2~3 つの軌跡において末尾の 3 手が共通する傾向にあることがわかった。

表 A.3: 再グループ化した際の軌跡の数

手数 (盤面数, 補間の有無)	再グループ化前の軌跡数 (平均)	再グループ化後の軌跡数 (平均)
19-24(9862, 無)	7.01	3.00
19-24(9862, 有)	13.62	4.43
13-24(21022, 無)	6.95	3.08
13-24(21022, 有)	12.93	4.51
0-40(61021, 無)	5.23	2.23
0-40(61021, 有)	10.82	3.78

また、各軌跡の長さの平均は A.4 のようになった。尚、正確には 1 つの盤面、行動 (最も訪問回数が大きい行動) の組に対して取り出される軌跡の平均を更に平均した値を示している。

表 A.4: 軌跡の長さの平均

手数 (盤面数, 補間の有無)	軌跡の長さ (平均)
0-40(61021, 無)	10.89
0-40(61021, 有)	15.31

## A.5 グレー ボックス的手法

モデルのパラメータを固定し、再び第 4 章のデータ実験を行った。パラメータの値を固定しているため、本実験はモデルの具体的なパラメータの値へのアクセスが不可能な場合にも適用可能なグレー ボックス的手法であると言える。固定されたパラメータを表 A.5 に示す。

表 A.5: データ実験 (追加): 使用モデルのパラメータ

パラメータ名	値
time	5
$C_{puct}$	1
エポック数	200
$k$ (提案手法のみ)	4
$l$ (提案手法のみ)	2

実験の結果は表 A.6 のようになった。第 4 章に記載したホワイトボックス的手法と同様に提案手法は group prediction rate において比較手法より高い値を示した。

表 A.6: 実験結果: データ実験

	group prediction rate	stone prediction rate		
手数 (盤面数, 補間の有無)	比較手法	提案手法	比較手法	提案手法
19-24(9862, 無)	0.44	<b>0.60</b>	<b>0.63</b>	0.61
19-24(9862, 有)	0.44	<b>0.63</b>	<b>0.63</b>	0.61
13-24(21022, 無)	0.37	<b>0.53</b>	<b>0.56</b>	0.55
13-24(21022, 有)	0.37	<b>0.55</b>	<b>0.56</b>	0.55

## 付録B

### 各種アルゴリズムの詳細

#### B.1 PV-モンテカルロ木探索

PV-モンテカルロ木探索の疑似コードを Algorithm1 , Algorithm2 に示す .

#### B.2 提案手法のアルゴリズム

提案手法のアルゴリズムの疑似コードを Algorithm3 , Algorithm4 に示す .

#### B.3 比較手法のアルゴリズム

比較手法の疑似コードを Algorithm 5 に示す . 疑似コード中の Traverse() は Algorithm4 と同一である .

#### B.4 提案手法におけるニューロン補間

第4章におけるデータ実験 , インタフェース実験ではいずれも手法のニューラルネットワークによる補間を行っている . 第4章で示した疑似コードでは未探索のノードにたどり着いた際は走査を終了する . ニューラルネットワークによる補間とはこの場合 , 方策  $P(s, a)$  で訪問回数  $N(s, a)$  を代用し走査を継続することである . Algorithm6 , Algorithm7 にニューロン補間を行う場合の疑似コー

---

**Algorithm 1** PV-MCTS in AlphaZero (Part 1: Exploration)

---

```
1:  $t$ : 決定木
2:  $T$ : 遷移関数
3:  $N(s, a)$ :  $(s, a)$  の組み合わせを探索した回数
4:  $Q(s, a)$ : 行動価値関数
5:  $W(s, a)$ : 行動価値の総和 ( $W(s, a) = Q(s, a)N(s, a)$ )
6:  $P(s, a)$  ( $= P(s_n), s_n = T(s, a)$ ):
7: ニューラルネットワークから出力された方策
8:  $V(s, a)$  ( $= V(s_n), s_n = T(s, a)$ ):
9: ニューラルネットワークから出力された局面評価
10: function EXPLORE( $s_{\text{start}}$ )
11:     Set  $s_{\text{now}} = s_{\text{start}}$  and  $a_{\text{now}} = a_m$ 
12:     for each simulation do
13:          $\zeta \leftarrow$  empty list
14:          $s_{\text{current}} \leftarrow s_{\text{start}}$ 
15:         while  $s_{\text{current}}$  がゲームの終了状態でない場合 do
16:              $a_t \leftarrow \text{TreePolicy}(s)$ 
17:             (T は遷移関数)
18:             ( $s_{\text{current}}, a_t$ ) を  $\zeta$  の末尾に追加
19:              $s_{\text{next}} \leftarrow T(s_{\text{current}}, a_t)$ 
20:              $s_{\text{current}} \leftarrow s_{\text{next}}$ 
21:         end while
22:          $G \leftarrow V(s_e)$ 
23:         ( $s_e$  は  $s_{\text{start}}$  から探索してたどり着いたノード  $s_e$ )
24:         BACKPROPAGATE( $\zeta, G$ )
25:     end for
26: end function
```

---

**Algorithm 2** PV-MCTS in AlphaZero (Part 2: Backpropagation)

---

```

1: function TREEPOLICY( $s$ )
2:   if  $s$  が探索されていない子ノードを持つとき then
3:      $s_c \leftarrow T(s, a)$  ( $s_c$  は未探索のノード)
4:     INITNODE( $s_c$ )
5:      $\blacktriangleleft_a$ 
6:   else
7:     以下の PUCT スコア  $U(s, a)$  を計算
8:     
$$U(s, a) = C(s)P(s, a) \frac{\sqrt{N(s)}}{1+N(s,a)}$$

9:     
$$C(s) = \log \frac{1+N(s)+C_{\text{base}}}{C_{\text{base}}} + C_{\text{init}}$$

10:    ( $C_{\text{base}}, C_{\text{init}}$  はハイパーパラメータ)
11:    ( $N(s) = \Gamma N(s, a)$ )
12:    以下のように  $a$  を求める
13:     $a = \operatorname{argmax}_a(Q(s, a) + U(s, a))$ 
14:     $\blacktriangleleft_a$ 
15:   end if
16: end function
17: function BACKPROPAGATE( $\zeta, G$ )
18:   for each node-action pair  $(s, a)$  in  $\zeta$  do
19:      $N(s, a) \leftarrow N(s, a) + 1$ 
20:      $W(s, a) \leftarrow W(s, a) + G$ 
21:      $Q(s, a) \leftarrow \frac{W(s,a)}{N(s,a)}$ 
22:   end for
23: end function
24: function INITNODE( $s$ )
25:   for each action  $a$  from  $s$  do
26:      $N(s, a) \leftarrow 0$ 
27:      $W(s, a) \leftarrow 0$ 
28:      $Q(s, a) \leftarrow 0$ 
29:   end for
30: end function

```

---

---

**Algorithm 3 提案手法のアルゴリズム (part1)**

---

1:  $t$ : 手法を適用する探索木  
2:  $T$ : 状態遷移関数  
3:  $l$ : 盤面の収集を行う手数  
4:  $k$ : 一度に集める盤面の数  
5:  $\zeta(s, s')$ : ノード  $s$  から  $s'$  までの軌跡  
6: **function** MYALGORITHM( $s_{start}, a, l, k$ )  
7:      $s_{now} \leftarrow s_{start}$   
8:      $a_{now} \leftarrow a$   
9:      $Z_0 \leftarrow \text{COLLECTBOARDS}(s_{now}, a_{now}, l, k)$   
10:    ( $Z_0 = \{\zeta(s_{start}, s'_{1}), \dots, \zeta(s_{start}, s'_{k^l})\}$ )  
11:     $Z \leftarrow \text{empty list}$   
12:    **for** each  $\zeta(s_{start}, s'_{i})$  in  $Z_0$  **do**  
13:        $\zeta(s_{start}, s_{edge_i}) \leftarrow \text{TRAVERSE}(s'_{i}, \zeta(s_{start}, s'_{i}))$   
14:        $\zeta(s_{start}, s_{edge_i})$  を  $Z$  の末尾に追加  
15:    **end for**  
16:    収集された終了状態の集合  $S (= \{s_{edge_1}, s_{edge_2}, \dots, s_{edge_{k^l}}\})$  を任意の共通  
項  $c$  で副集合  $\{S_1, S_2, \dots, S_q\}$  に分割する  
17:    最も要素数の多い副集合  $S_{max}$  中の各要素  $\{s_{e_1}, s_{e_2}, \dots, s_{e_u}\}$   
18:    に対応する軌跡の集合  $Z_{max} (= \{\zeta(s_{start}, s_{e_1}), \zeta(s_{start}, s_{e_2}), \dots, \zeta(s_{start}, s_{e_u})\})$   
    を保存  
19:    **return**  $Z_{max}$   
20: **end function**

---

**Algorithm 4 提案手法のアルゴリズム (part2)**

---

```

1: function COLLECTBOARDS( $s, a, l, k$ )
2:    $s_{now} \leftarrow T(s, a)$ 
3:    $Z \leftarrow$  empty queue
4:   if  $s$  が未探索のノード ( $N(s) = 0$ ) または終了状態のとき then  $\leftarrow Z$ 
5:   end if
6:   訪問回数  $N(s)$  から上位  $k$  の行動  $\{a_1, a_2, \dots, a_k\} (= \alpha)$  を取り出す
7:   for each  $a_i$  in  $\alpha$  do
8:      $s_{next_i} \leftarrow T(s_{now}, a_i)$ 
9:      $\zeta(s_{now}, s_{next_i}) (= \{s_{now}, a_i, s_{next_i}\})$  を  $Z$  の末尾に追加
10:    end for
11:   if  $l=1$  then  $\leftarrow Z$ 
12:   end if
13:    $i \leftarrow 1$ 
14:   while  $i < l$  do
15:     for each  $\zeta(s_{now}, s_j)$  in  $Z$  do
16:        $\zeta(s_{now}, s_j)$  を  $Z$  からポップ
17:       if  $s$  が未探索のノード ( $N(s) = 0$ ) または終了状態のとき then
18:          $\zeta(s_{now}, s_j)$  を  $k$  回  $Z$  の末尾に追加
19:         continue
20:       end if
21:       訪問回数  $N(s)$  から上位  $k$  の行動  $\{a_1, a_2, \dots, a_k\} (= \alpha)$  を取り出す
22:       for each  $a_i$  in  $\alpha$  do
23:          $s_{next_j} \leftarrow T(s_j, a_i)$ 
24:          $\zeta(s_{now}, s_{next_i}) (= \zeta(s_{now}, s_j).append(a_i, s_{next_i}))$  を  $Z$  の末尾に追加
25:       end for
26:     end for
27:   end while  $\leftarrow Z (= \{\zeta(s_{start}, s'_{1}), \dots, \zeta(s_{start}, s'_{k^l})\})$ 
28: end function

29: function TRAVERSE( $s, \zeta(s_{start}, s)$ )
30:    $s_{now} \leftarrow s$ 
31:    $\zeta_r \leftarrow \zeta(s_{start}, s)$ 
32:   while  $s_{now}$  が探索済みかつ終了状態でない do
33:      $a_t \leftarrow \text{argmax}_a N(s_{now}, a)$ 
34:      $s_n \leftarrow T(s_{now}, a_t)$ 
35:      $\zeta_r.append(a_t, s_n)$ 
36:      $s_{now} \leftarrow s_n$ 
37:   end while  $\leftarrow \zeta_r$ 
38: end function

```

---

---

**Algorithm 5 比較手法のアルゴリズム**

---

```
1:  $t$ : 手法を適用する探索木
2:  $T$ : 状態遷移関数
3:  $\zeta(s, s')$ : ノード  $s$  から  $s'$  までの軌跡
4: function COMPAREALGORITHM( $s, a$ )
5:    $s_n \leftarrow T(s, a)$ 
6:    $\zeta(s, s_n) \leftarrow \{s, a, s_n\}$ 
7:    $\zeta \leftarrow s_n, \zeta(s, s_n)(\leftarrow) \zeta$ 
8: end function
9: function TRAVERSE( $s, \zeta(s_{start}, s)$ )
10:    $s_{now} \leftarrow s$ 
11:    $\zeta_r \leftarrow \zeta(s_{start}, s)$ 
12:   while  $s_{now}$  が探索済みかつ終了状態でない do
13:      $a_t \leftarrow \text{argmax}_a N(s_{now}, a)$ 
14:      $s_n \leftarrow T(s_{now}, a_t)$ 
15:      $\zeta_r.append(a_t, s_n)$ 
16:      $s_{now} \leftarrow s_n$ 
17:   end while  $\zeta_r$ 
18: end function
```

---

ドを示す（変更部分に下線）．比較手法に対してニューロ補間を行う場合も同様に Traverse() のコードを変更する．

## B.5 インタフェース実験における提案手法の変更

インタフェース実験では画面の左下に盤面  $s$  の訪問回数  $N(s)$  と局面評価  $V(s, a)$  を表示する．このとき  $V(s, a)$  の絶対値（どちらのプレイヤーの勝利に近いか）と予想図の勝敗が異なる場合，ユーザーの混乱を招く可能性がある．そのため，提案手法では勝敗が  $V(s, a)$  の絶対値と一致する軌跡を表示するように修正を施した．インタフェース実験用に修正された疑似コードは Algorithm:8 となる（下線が変更部分）また，提案手法により収集した最終盤面  $S = \{s_{edge_1}, s_{edge_2}, \dots, s_{edge_{kl}}\}$  は最終的に 4 つ以上連続してつながっている石の組み合わせ（fatal group）でグループ化し，最も要素数が多い集合の軌跡をユーザーに提示した．

**Algorithm 6 提案手法のアルゴリズム (ニューロン補間あり)part1**

---

```

1: function COLLECTBOARDS( $s, a, l, k$ )
2:    $s_{now} \leftarrow T(s, a)$ 
3:    $Z \leftarrow$  empty queue
4:   if  $s$  が終了状態のとき then  $\leftarrow Z$ 
5:   end if
6:   if  $s$  が未探索のとき then
7:     方策  $P(s)$  から上位  $k$  の行動  $\{a_1, a_2, \dots, a_k\} (= \alpha)$  を取り出す
8:   else
9:     訪問回数  $N(s)$  から上位  $k$  の行動  $\{a_1, a_2, \dots, a_k\} (= \alpha)$  を取り出す
10:  end if
11:  for each  $a_i$  in  $\alpha$  do
12:     $s_{next_i} \leftarrow T(s_{now}, a_i)$ 
13:     $\zeta(s_{now}, s_{next_i}) (= \{s_{now}, a_i, s_{next_i}\})$  を  $Z$  の末尾に追加
14:  end for
15:  if  $l=1$  then  $\leftarrow Z$ 
16:  end if
17:   $i \leftarrow 1$ 
18:  while  $i < l$  do
19:    for each  $\zeta(s_{now}, s_j)$  in  $Z$  do
20:       $\zeta(s_{now}, s_j)$  を  $Z$  からポップ
21:      if  $s$  終了状態のとき then
22:         $\zeta(s_{now}, s_j)$  を  $k$  回  $Z$  の末尾に追加
23:        continue
24:      end if
25:      if  $s$  が未探索のとき then
26:        方策  $P(s)$  から上位  $k$  の行動  $\{a_1, a_2, \dots, a_k\} (= \alpha)$  を取り出す
27:      else
28:        訪問回数  $N(s)$  から上位  $k$  の行動  $\{a_1, a_2, \dots, a_k\} (= \alpha)$  を取り出す
29:      end if
30:      for each  $a_i$  in  $\alpha$  do
31:         $s_{next_j} \leftarrow T(s_j, a_i)$ 
32:         $\zeta(s_{now}, s_{next_i}) (= \zeta(s_{now}, s_j).append(a_i, s_{next_i}))$  を  $Z$  の末尾に追加
33:      end for
34:    end for
35:  end while  $\leftarrow Z (= \{\zeta(s_{start}, s'_{1}), \dots, \zeta(s_{start}, s'_{k^l})\})$ 
36: end function

```

---

---

Algorithm 7 提案手法のアルゴリズム(ニューロ補間あり)part2

---

```
1: function TRAVERSE( $s, \zeta(s_{start}, s)$ )
2:    $s_{now} \leftarrow s$ 
3:    $\zeta_r \leftarrow \zeta(s_{start}, s)$ 
4:   while  $s_{now}$  が終了状態でない do
5:     if  $s_{now}$  が未探索のとき then
6:        $a_t \leftarrow \text{argmax}_a P(s_{now}, a)$ 
7:     else
8:        $a_t \leftarrow \text{argmax}_a N(s_{now}, a)$ 
9:     end if
10:     $s_n \leftarrow T(s_{now}, a_t)$ 
11:     $\zeta_r.append(a_t, s_n)$ 
12:     $s_{now} \leftarrow s_n$ 
13:   end while  $\zeta_r$ 
14: end function
```

---

---

**Algorithm 8 提案手法のアルゴリズム (インターフェース実験)**

---

```
1: function TRAVERSE( $s, \zeta(s_{start}, s)$ )
2:    $\zeta(s_{start}, s)$  から  $s_{start}$  の次の行動  $a$  を取り出す
3:    $v \leftarrow V(s, a)$ 
4:    $s_{now} \leftarrow s$ 
5:    $\zeta_r \leftarrow \zeta(s_{start}, s)$ 
6:   while  $s_{now}$  が終了状態でない do
7:     if  $s_{now}$  が未探索のとき then
8:        $a_t \leftarrow \text{argmax}_a P(s_{now}, a)$ 
9:     else
10:       $a_t \leftarrow \text{argmax}_a N(s_{now}, a)$ 
11:    end if
12:     $s_n \leftarrow T(s_{now}, a_t)$ 
13:     $\zeta_r.append(a_t, s_n)$ 
14:     $s_{now} \leftarrow s_n$ 
15:  end while
16:  if  $v$  と  $V(s_{now})$  の絶対値が異なるとき then  $\leftarrow$  null
17:  end if  $\leftarrow$   $\zeta_r$ 
18: end function
```

---

## 付録 C

### alphazero\_baseline

#### C.1 ニューラルネットワーク

モデルの構成を図 C.1 に示す。alphazero\_baseline における選択肢は列の数と等しい 7 であるため、 $1 \times 7$  の行列となる。例えば方策が  $\{0, 0.1, 0.2, 0, 0, 0.7, 0.8\}$  であるとき、方策中の最も大きい成分は 7 番目の 0.8 であるため、プレイヤーは次に 7 列目を選択する事が推奨される。また、モデルの訓練過程は以下のステップをまとめて 1 エポックとした繰り返しによって構成されている。使用したテストデータ [52] は - 木探索による connect4 の解から生成されている [53]。

1. 500 ゲーム分の自己対戦を行う
2. 自己対戦によって収集した盤面を入力としてネットワークを訓練
3. ネットワークを教師データによって評価
4. 新しいネットワークを用いた AI と訓練前の最善のネットワークを用いた AI による対戦を 50 ゲーム分行い 16 割以上の勝率を記録した場合、新しいネットワーク最善のネットワークとして保存する

本論文の第 3 章におけるインターフェース実験で用いた「強い AI」のニューラルネットワークは上記をステップを 200 エポック分実行して訓練されたモデルを使用している。

また、ネットワーク訓練時のパラメータの値を C.1 に示す。

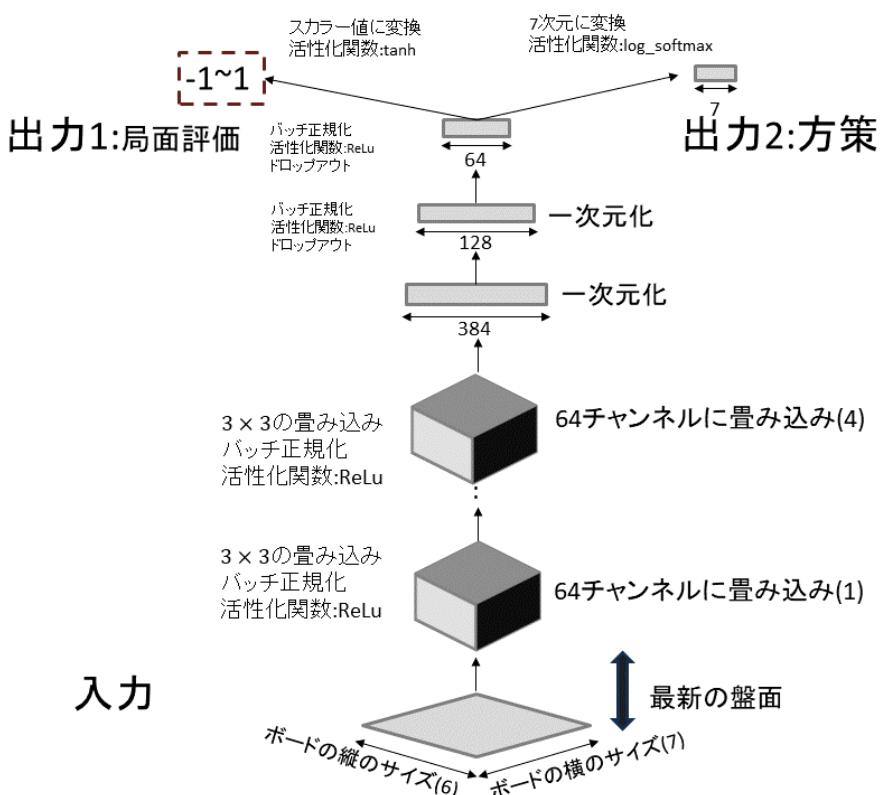


図 C.1: alphazero\_baseline ネットワークの構成

表 C.1: ネットワーク訓練時のパラメータ名と値

パラメータ名	値
学習率 (lr)	0.001
dropout	0.3
epochs	5
batch_size	64
num_channels	64

## C.2 alphazero\_baseline のパラメータ更新

alphazero\_baseline のパラメータ更新は第 2 章で述べた手順とほぼ同一であるが PUCT スコア  $U(s, a)$  の定義と  $Q(s, a)$  の更新則に差異がある。 $C_{\text{cpuct}}$  はハイパー-パラメータである。 $(N(s), N(s, a))$  はそれぞれ  $s, (s, a)$  に対して探索を行った回数)

$$U(s, a) = C_{\text{cpuct}} P(s, a) \frac{\sqrt{N(s)}}{1 + N(s, a)} \quad (\text{C.1})$$

$Q(s, a)$  は以下のように更新される。 $(s_c)$  は現在の探索ノード  $s$  から見た最も PUCT スコア  $U(s, a)$  と  $Q(s, a)$  の和の高い子ノード)

$$Q(s, a) = \frac{N(s, a)Q(s, a) + V(s_c)}{N(s, a) + 1} \quad (\text{C.2})$$

これらの変更を加えた alphazero\_baseline における PV-MCTS のアルゴリズムを Algorithm9 に示す。

**Algorithm 9** PV-MCTS in alphazero-baseline (変更部分)

---

```
1:  $t$ : 決定木
2:  $T$ : 遷移関数
3:  $N(s, a)$ :  $(s, a)$  の組み合わせを探索した回数
4:  $Q(s, a)$ : 行動価値関数 ( $\text{Explore}(s)$  の平均)
5:  $W(s, a)$ : 行動価値の総和 ( $W(s, a) = Q(s, a)N(s, a)$ )
6:  $P(s, a)$  ( $= P(s_n), s_n = T(s, a)$ ):
7: ニューラルネットワークから出力された方策
8:  $V(s, a)$  ( $= V(s_n), s_n = T(s, a)$ ):
9: ニューラルネットワークから出力された局面評価
10: function TREEPOLICY( $s$ )
11:   if  $s$  が探索されていない子ノードを持つとき then
12:      $s_c \leftarrow T(s, a)$  ( $s_c$  は未探索のノード)
13:     INITNODE( $s_c$ )
14:     ↙ $a$ 
15:   else
16:     以下の PUCT スコア  $U(s, a)$  を計算
17:     
$$U(s, a) = C_{\text{cpuct}} P(s, a) \frac{\sqrt{N(s)}}{1+N(s,a)}$$

18:     ( $N(s) = \Gamma N(s, a)$ )
19:     以下のように  $a$  を求める
20:      $a = \text{argmax}_a(Q(s, a) + U(s, a))$ 
21:     ↙ $a$ 
22:   end if
23: end function
24: function BACKPROPAGATE( $\zeta, G$ )
25:   for each node-action pair  $(s, a)$  in  $\zeta$  do
26:      $N(s, a) \leftarrow 0$ 
27:      $W(s, a) \leftarrow 0$ 
28:      $Q(s, a) \leftarrow 0$ 
29:   end for
30: end function
31: function INITNODE( $s$ )
32:   for each action  $a$  from  $s$  do
33:      $N(s, a) \leftarrow N(s, a) + 1$ 
34:      $W(s, a) \leftarrow W(s, a) + G$ 
35:     
$$Q(s, a) \leftarrow \frac{N(s,a)Q(s,a)+V(s_c)}{N(s,a)+1}$$

36:     ( $s_c = T(s, a)$ )
37:   end for
38: end function
```

---

## 付録D

### インタフェース実験の詳細

インタフェース実験では全3回にわたる実験を行い、被験者同士の対戦を行う第3回以外の2回は被験者がAIとの対戦とAIを用いた振り返りを行う。第4章と同様に3回セットの実験のうち、1回目と2回目を第1段階、3回目を第2段階と呼ぶ。以下で被験者のデータの詳細を述べ、それから実験設定や結果の詳細を段階ごとに述べていく。

#### D.1 被験者のデータ

実験対象者は計22人の10代-20代学生(男性17名:女性5名)であり、表D.1に示す各種の事前質問(1~5の値で答える)の結果は以下となった。ボードゲームの経験を問う質問に関しては1を「1年に1回程プレイする」程度、5を「週に2,3回以上プレイする」程度と定めた。connect4の経験を問う質問に関しては1を「connect4を知らない」程度、5を「週に2,3回以上プレイする」程度と定めた。また、機械学習と強化学習の知識を問う質問に関しては1を「言葉を聞いたことがある」程度、5を「機械学習についての入門書を読んだり、授業を受けたことがある」程度と定めた。また、被験者には先述のAllis[49]による9strategyをはじめとしたconnect4の知識的解法をネット検索等の手段で調べないよう要請した。

表 D.1: 事前質問項目:インターフェース実験

質問	回答の平均
ボードゲームの経験はどれくらいありますか	2.18
connect4(ゲーム)の経験はどれくらいありますか	1.64
機械学習に対する知識はどれくらいありますか	3.59
強化学習に対する知識はどれくらいありますか	3.23

## D.2 第1段階

第1段階はAIとの対戦とその振り返りによって構成されている。振り返りモードでは表示する進行図の手数、AIの強さを被験者ごとに変更しつつ実験を行った。進行図の手数は{3, 5, 7, 制限なし(最後まで表示)}の4段階、AIの強さは{強, 弱}の2段階を設定した。また、対戦を行うAIモデルを用いて振り返りの進行図を生成した。使用したAIモデル(alphazero\_baseline)、提案手法に与えるパラメータを表D.2に示す。

表 D.2: AIモデルのパラメータ(インターフェース実験第1段階)

モデル	強	弱
time	5	1
$C_{puct}$	1	0.25
エポック数	200	1
$k$ (提案手法のみ)	4	4
$l$ (提案手法のみ)	2	2

また、1回の実験当たりの制限時間は7分とし、実験内のAIとの対戦は2回までに制限した。これは実戦経験による熟達ではなく支援システムによる学習過程の観察という実験の趣旨に従うためである。

### D.2.1 振り返りモード(GUI)の詳細

振り返りモードでは直前の対戦の内容を振り返ることができる「*左/右*」ボタンを押すことで「1つ先の盤面に進む/戻る」ことができる。また、振り返りを補助する機能として画面左下の「counts」に表示されている盤面  $s$  に対する各行動の訪問回数  $N(s)$ 、「value」に局面評価  $V(s)$  を示した。 $N(s), V(s)$  は対戦に使用したモデルから導かれる。数字が書かれているボタンは AI モデル(対戦に使ったモデルと同じ)による進行図を提示するボタンであり、ボタンに描かれた数字と同じ列を選択した場合の進行図を見る能够である。図 D.1 では「4」と描かれたボタンを押した結果、左から 4 番目を選択した場合の進行図が表示されていることがわかる。選択した位置は「\*」マークで表示される。

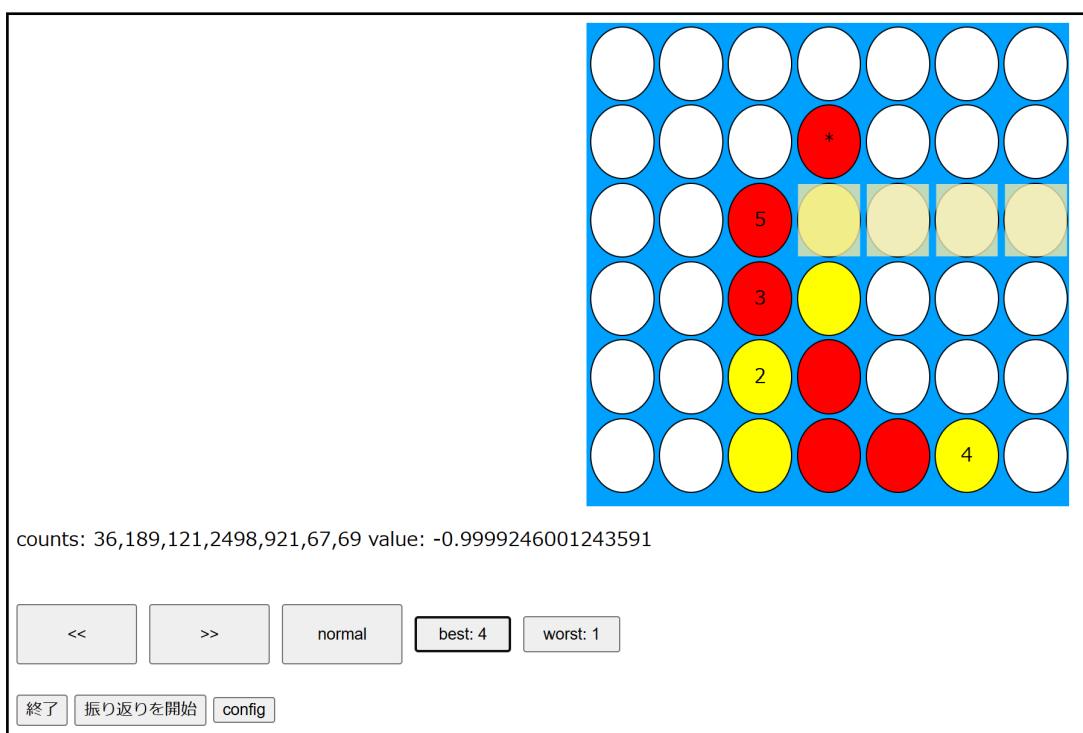
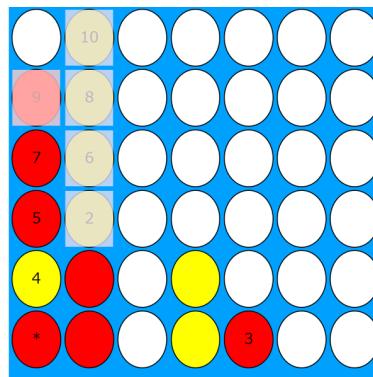


図 D.1: 進行図の提示

想定図中の数字は数字の大きさの順番に石が置かれるという AI の予想を表している。図 D.2 では赤が「\*」マークの位置に石を置いた次に黄は左から 3 列目の位置、その次に赤は左から 3 列目に打つことが予測されている。また、図中で

色づけられている部分は進行図においてその部分がは4つ以上つながることを示している。赤く色づけられている場合はその部分で赤が4つ以上つながると予想されていること、黄色く色づけられている場合はその部分で黄色が4つ以上つながると予想されていることを示している。また、提案手法による振り返りが適用されたグループでは提示される進行図が複数となる。そのため図D.2に示すように他の想定図を見る能够性がある「next\_traj/pre\_traj」ボタンが追加される。ここからは第1段階中の1日目、2日目のGUIの違いについて述べる。



4 | 5 | 6 | 7 | next\_traj | pre\_traj | close

図 D.2: 提案手法による進行図の提示

### D.2.2 1日目

1日目は被験者の操作への慣れを促進するため全ての選択肢に対し進行図表示ボタン（数字ボタン）を設置した。図D.3に示すようにこれらのボタンは「traj」ボタンを押すことで出現する。

### D.2.3 2日目

2日目は進行図の閲覧を促進するため traj ボタンを廃止し、最も訪問回数  $N(s, a)$  の多い選択肢  $a_{max}$ 、少ない選択肢  $a_{min}$  からの進行図を示すボタンを

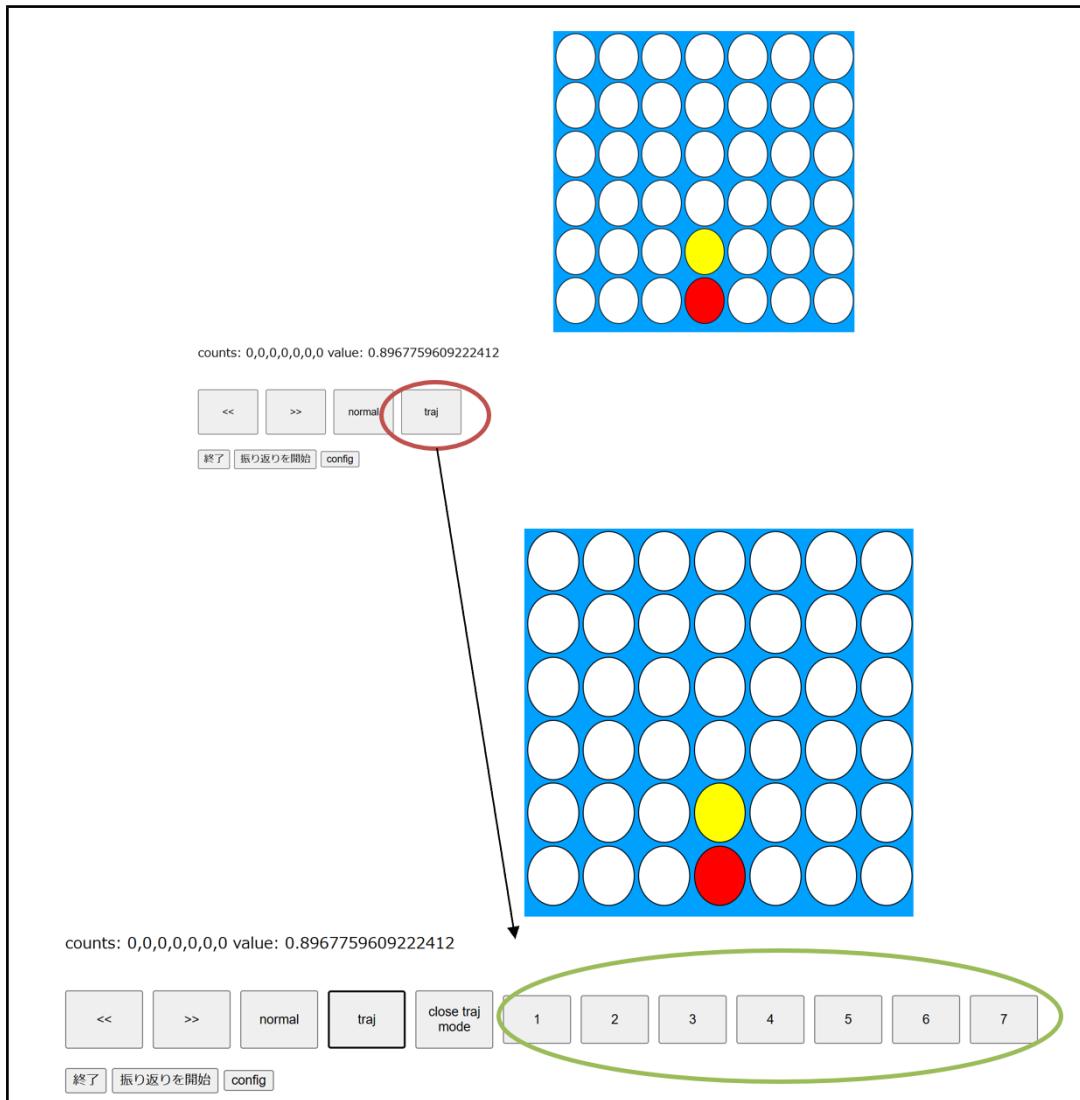


図 D.3: 進行図表示ボタン (1 日目)

それぞれ「best:(列の数字)」「worst:(列の数字)」のラベルで図 D.4 のように表示した。

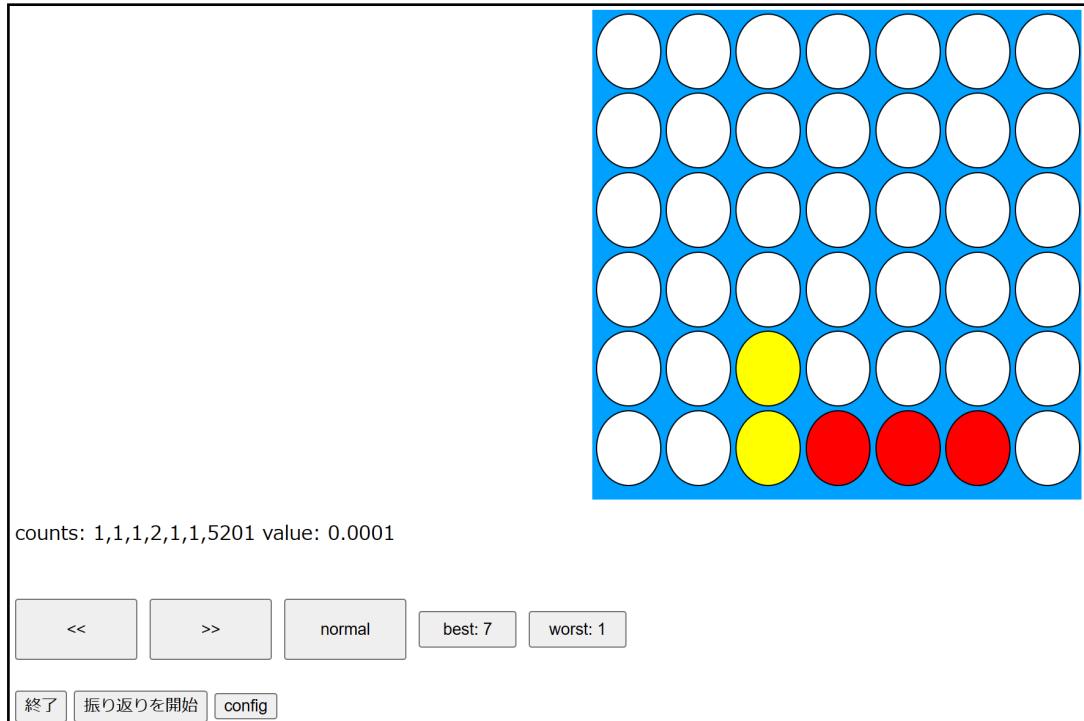


図 D.4: 進行図表示ボタン (2 日目)

### D.3 第2段階

第2段階は提案手法を用いて振り返りを行ったグループ(A)の被験者と比較手法を用いて振り返りを行ったグループ(B)の被験者の対戦である。1回の実験の中で2人の被験者が手番を入れ替え2回の対戦を行った。制限時間は1ゲーム5分であり、制限時間内に終わらなかったゲームはAIによって判定した。具体的には制限時間が切れた際の盤面  $s$  のAIによる局面評価  $V(s)$  の絶対値が0.5以上である場合にはその符号が示すプレイヤーの勝利とし、 $V(s)$  の絶対値が0.5未満である場合は引き分けとする。判定用に使用したAIモデルのパラメータはD.3に示す。

表 D.3: AI モデルのパラメータ (インターフェース実験第 2 段階)

パラメータ名	値
time	0
$C_{puct}$	1
エポック数	200

## D.4 質問項目の詳細

質問項目は「タスクの熟達度に関する質問」((a)) と「タスクの楽しさや面白さに関する質問」((b)) の 2 つに分けられる。具体的な質問項目は D.4 に記載した。また、2 日目は「タスクの楽しさや面白さに関する質問」に「1 回目と比べて成長したと感じますか」を追加した。

表 D.4: 質問項目: インタフェース実験

質問の種類	質問の内容
タスクの熟達度に関する質問	振り返りによって成長したと感じましたか 振り返りによって AI の意図を掴むことができたと感じますか (2 日目のみ) 1 回目と比べて成長したと感じますか
タスクの楽しさや面白さに関する質問	今後も connect4 をプレイするときにこのシステムを使いたいと思いましたか システムにより振り返りが楽しくなったと感じますか

## D.5 結果データ

第 4 章では先読みの手数ごとに結果データを集計した。ここでは他の観点から集計されたデータを示す。表 D.5 に先読みの手数を分けずに合計した結果を示す。

表 D.5: 結果: 総合

質問の種類	質問の内容	主観評価	
		B(比較手法)	A(提案手法)
(a)	振り返りによって成長したと感じましたか	3.86	3.55
	振り返りによって AI の意図を掴むことができたと感じますか	3.32	2.73
	(2 回目のみ) 1 回目と比べて成長したと感じますか	4.00	3.36
(b)	今後も connect4 をプレイするときにこのシステムを使いたいと思いましたか	4.27	4.05
	システムにより振り返りが楽しくなったと感じますか	4.33	3.86

また，AIの強さごとに集計した結果を表D.6，表D.7に示す。表D.6に強いAIを用いて訓練した被験者のデータ，表D.7に弱いAIを用いて訓練した被験者のデータを示す。t検定で有意な差を示した箇所に「\*」印を付与している。

表 D.6: 結果:AI の強さ (強い)

質問		主観評価	
質問の種類	質問の内容	B(比較手法)	A(提案手法)
(a)	振り返りによって成長したと感じましたか	<b>3.82</b>	3.67
	振り返りによって AI の意図を掴むことができたと感じますか	<b>3.55*</b>	2.58*
	(2回目のみ) 1回目に比べて成長したと感じますか	<b>4.00</b>	3.00
(b)	今後も connect4 をプレイするときにこのシステムを使いたいと思いましたか	<b>4.64*</b>	4.08*
	システムにより振り返りが楽しくなったと感じますか	<b>4.50</b>	3.75

表 D.7: 結果:AI の強さ (弱い)

質問		主観評価	
質問の種類	質問の内容	B(比較手法)	A(提案手法)
(a)	振り返りによって成長したと感じましたか	<b>3.80</b>	3.33
	振り返りによって AI の意図を掴むことができたと感じますか	3.00	3.00
	(2回目のみ) 1回目に比べて成長したと感じますか	<b>4.00</b>	3.80
(b)	今後も connect4 をプレイするときにこのシステムを使いたいと思いましたか	3.80	<b>4.00</b>
	システムにより振り返りが楽しくなったと感じますか	<b>4.1</b>	4

また，1日目のデータを表D.8，2日目のデータを表D.9に記載する。t検定で有意な差を示した箇所に「\*」印を付与している。

表 D.8: 結果:1日目

質問		主観評価	
質問の種類	質問の内容	B(比較手法)	A(提案手法)
(a)	振り返りによって成長したと感じましたか	<b>3.78</b>	3.56
	振り返りによって AI の意図を掴むことができたと感じますか	2.94	<b>3.05</b>
(b)	今後も connect4 をプレイするときにこのシステムを使いたいと思いましたか	<b>4.28</b>	4.06
	システムにより振り返りが楽しくなったと感じますか	<b>4.28</b>	4.06

表 D.9: 結果:2日目

質問		主観評価	
質問の種類	質問の内容	B(比較手法)	A(提案手法)
(a)	振り返りによって成長したと感じましたか	<b>3.91</b>	3.55
	振り返りによって AI の意図を掴むことができたと感じますか	<b>3.64*</b>	2.45*
	(2回目のみ) 1回目に比べて成長したと感じますか	<b>4.00</b>	3.36
(b)	今後も connect4 をプレイするときにこのシステムを使いたいと思いましたか	<b>4.18</b>	4.00
	システムにより振り返りが楽しくなったと感じますか	<b>4.40</b>	3.91

## 付録E

### 追加実験の詳細

インタフェース実験では全 3 回にわたる実験を行ったが追加実験ではシステムの実験の第 1 段階における 1 日目のみを実施した .

#### E.1 被験者のデータ

実験対象者は計 22 人の大学生 , 大学院生 (男性 6 名:女性 4 名) であり , インタフェース実験と同様の事前質問を行った結果を表 E.1 に示す .

表 E.1: 事前質問項目:追加実験

質問	回答の平均
ボードゲームの経験はどれくらいありますか	2.40
connect4(ゲーム) の経験はどれくらいありますか	1.60
機械学習に対する知識はどれくらいありますか	3.90
強化学習に対する知識はどれくらいありますか	3.40

#### E.2 実験設定の詳細

AI の強さは表 D.2 における「弱い AI」モデルを使用した . また , インタフェース実験における第 1 段階と同様に 1 回の実験当たりの制限時間は 7 分とし , 実験内の AI との対戦は 2 回までに制限した .

### E.2.1 振り返りモード (GUI) の詳細

GUI は主にインターフェース実験第 1 段階における 2 日目の GUI と類似している。ただし、図 E.1 に示すように訪問回数「counts」の表記を廃止し、「value」と数字ボタンは「traj&value」ボタンを押した際に出現する。

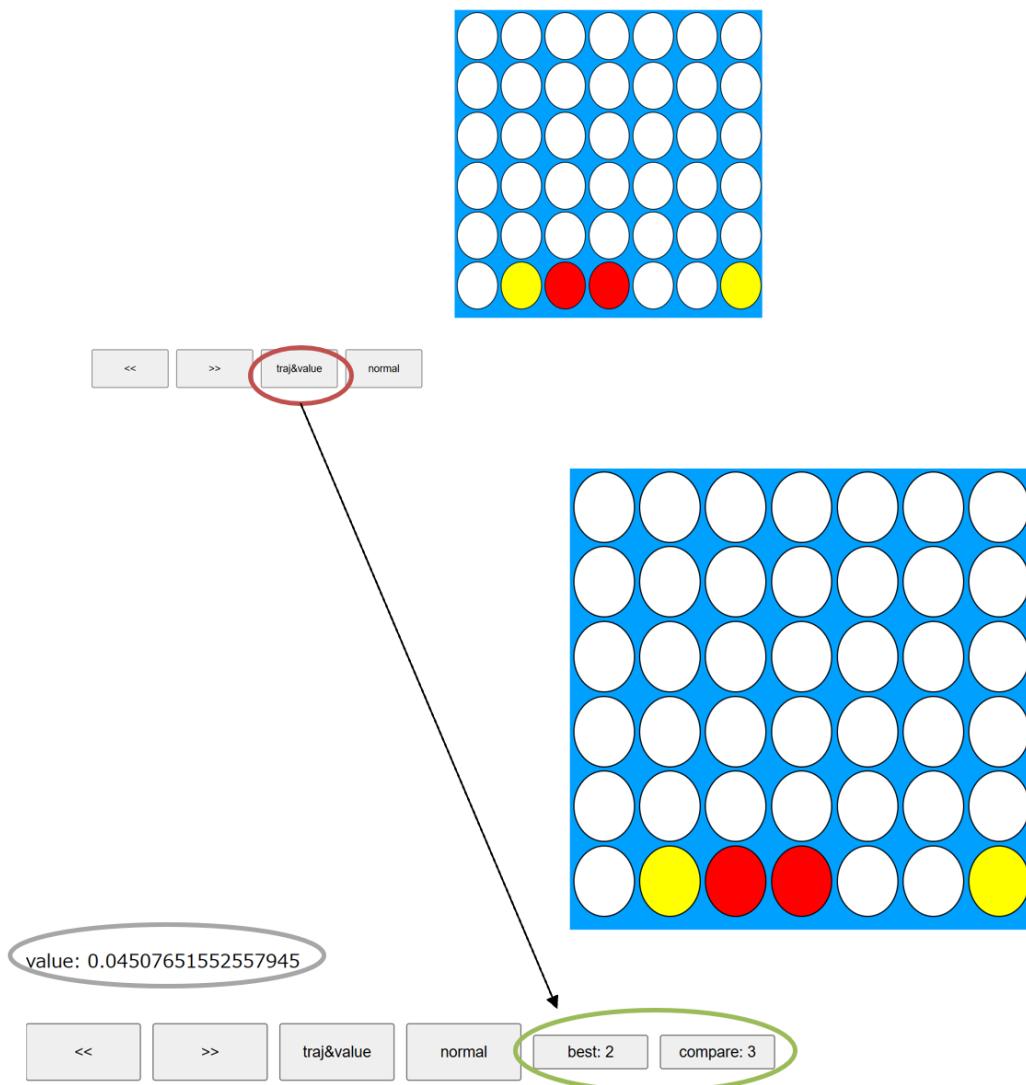


図 E.1: GUI(追加実験)

### E.2.2 アルゴリズムの変更

追加実験では最も訪問回数  $N(s, a)$  が大きい行動  $a_m$  から派生する進行図と  $a_m$  から「最も遠い」行動  $a_l$  から派生する進行図とを表示する「最も遠い」行動  $a_l$  は以下の手順によって定められる。また、アルゴリズムの疑似コードは Algorithm に記載している。

1. 盤面  $s$  から派生する各行動  $\{a_1, a_2, \dots, a_n\}$  に対して提案手法または比較手法による軌跡を収集する。その際に各行動に対して局面評価  $Q(s, a)$  と軌跡の長さの平均  $L(s, a)$  を記録する。
2. 最も訪問回数  $N(s, a)$  が大きい行動  $a_m$  による局面評価を  $v (= Q(s, a_m))$  とする。また、任意の閾値  $l$  に対して  $\text{margin}_m (= l - L(s, a_m))$  を計算する。
3. 局面評価の符号が  $v$  と異なる行動  $\{a_{d_1}, a_{d_2}, \dots, a_{d_k}\}$  を取り出す。
4.  $\{a_{d_1}, a_{d_2}, \dots, a_{d_k}\}$  内の各行動に対して  $\text{margin} (= l - L(s, a))$  を計算し、 $\text{margin}$  の符号が  $\text{margin}_m$  と異なる行動  $\{a'_1, a'_2, \dots, a'_k\}$  を取り出す。
5. 取り出した行動のうち軌跡の長さの平均  $L(s, a)$  が  $L(s, a_m)$  と最も遠い行動  $a_l$  を最適な行動  $a_m$  から最も遠い行動とする。

このアルゴリズムは状態  $s$ 、最適な行動  $a_m$  から派生する局面評価  $Q(s, a_m)$  と予測される軌跡の長さの平均  $L(s, a_m)$  が遠い行動を取り出している。つまり、「予測される勝敗」と「あとどのくらいでゲームが終了するか」の予想がなるべく  $a_m$  による予想と異なるものを取り出すことを意味している。

また、第 4 章の図 A.2 に記載したように、被験者に与える情報を可能な限り少なくする目的で取り出した進行図の軌跡を末尾の 3 手でさらにグループ化し、提案手法において 1 組の状態  $s$ 、行動  $a$  に対して閲覧できる進行図の数は 3 つまでとした。図 E.2 に再度、再グループ化の様子を示す。

**Algorithm 10** 追加実験のアルゴリズム

---

```
1:  $t$ : 手法を適用する探索木
2:  $T$ : 状態遷移関数
3:  $l$ : 任意の閾値
4:  $L(s, a)$ :  $(s, a)$  の進行図における軌跡の長さの平均
5: function EXTRACTDISTANTCHOICE( $s_{start}$ )
6:   for each  $i$  in  $\{1, 2, 3, 4\}$  do  $A_i \leftarrow$  empty list
7:   end for
8:    $a_m = \text{argmax}_a N(s_{start}, a)$ 
9:    $Z_m \leftarrow \text{MYALGORITHM}(s_{start}, a)$ 
10:   $L_m \leftarrow Z_m$  内の各軌跡の長さの平均
11:  margin  $\leftarrow (l - L_m)$ 
12:   $v_m \leftarrow Q(s, a_m)$ 
13:   $A \leftarrow \{a_1, a_2, \dots, a_n\}$  ( $A$  は  $s_{start}$  から派生する行動の集合)
14:  for each  $a_i$  in  $A$  do
15:     $Z_i \leftarrow \text{MYALGORITHM}(s_{start}, a_i)$ 
16:     $L_i \leftarrow Z_i$  内の各軌跡の長さの平均
17:    margin $_i \leftarrow (l - L_i)$ 
18:    if  $Q(s, a_i) \times v_m < 0$  then
19:      if margin  $\times$  margin $_i < 0$  then
20:         $A_1$  に  $a_i$  を追加
21:      else
22:         $A_2$  に  $a_i$  を追加
23:      end if
24:    else
25:      if margin  $\times$  margin $_i < 0$  then
26:         $A_3$  に  $a_i$  を追加
27:      else
28:         $A_4$  に  $a_i$  を追加
29:      end if
30:    end if
31:  end for
32:  for each  $i$  in  $\{1, 2, 3, 4\}$  do
33:    if  $A_i$  の要素数が 1 以上 then  $A_0 \leftarrow A_i$ 
34:    end if
35:  end for
36:   $d \leftarrow 0$ 
37:   $a_l \leftarrow \text{argmax}_a (|L(s, a) - L_m|)$ 
38:  ↪  $a_l$ 
39: end function
```

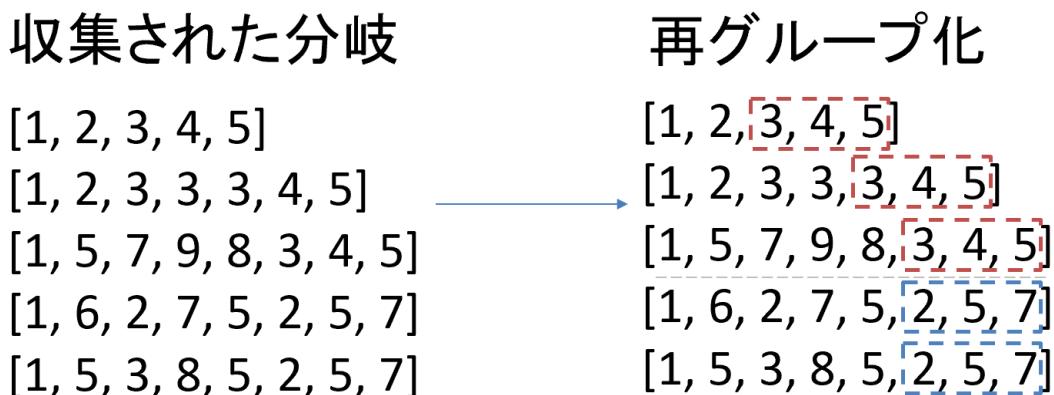


図 E.2: 再グループ化 (再掲)

### E.3 質問項目の詳細

質問項目は表 D.4 に記載した質問に加え、「タスクの熟達度に関連する質問 (a)」として「振り返りでは対戦を振り返るのに十分な量の情報を受け取ることができたと思いますか」という質問を追加した。