部分問題の解決を目的としたAlphaZero型強化学習の拡張

村上 花恋[1,a)]　金子 知適[1,b)]

概要：本稿では AlphaZero 型の深層強化学習を拡張し、囲碁において、対局だけでなく詰碁のように局面の一部分に注目してプレイする能力を訓練する手法を提案する。そのために、ネットワークの入力に注目範囲をあらわすチャネルを加え、指定された場合は対応する部分での利益を最大化するような方策や価値を出力するように設計する。そのような訓練が可能となるよう、自己対戦での棋譜は、複数の範囲を指示したものと、範囲の指示のない通常の対局の両方を用意する。Gumbel AlphaZero をベースに提案手法を実装し、九路盤の囲碁で性能評価したところ、100 世代訓練したエージェントは、指定された隅に概ね対応する出力ができていることを確認した。通常の対局の強さへの影響は、許容範囲であった。

Extension to AlphaZero-style reinforcement learning for solving life and death problem in Go

Murakami Karen[1,a)]　Tomoyuki Kaneko[1,b)]

Abstract: This paper extends AlphaZero-style deep reinforcement learning in Go to make agents understand strategies focusing on a specified region, or Tsumego problems, while keeping playing performance in the original game. To this end, our neural networks take a set of vertices as an additional input channel and yield a policy and value to achieve a best result in the specified region if it is not empty. To train such networks, self-play is also extended so that games with various regions regions and without regions are available for Tsumego training and ordinary training, respectively. We implemented our method based on Gumbel AlphaZero in nine by nine Go and empirically showed that our agents after 100 generation yield a decent policy for a specified corner with a slight drawback in usual playing performance.

1. はじめに

既存の AlphaZero 型強化学習 [1], [6], [8] はゲーム全体の勝利を目的にエージェントを訓練し、高い性能を発揮している。しかし、それらの学習方式は局所的な問題を解決できるとは限らない。本論文では AlphaZero 型モデルに対して入力に注目範囲を加えることで、囲碁における局所問題（詰碁）にニューラルネットワークである程度対応できるようにすることを目指す。

囲碁は、対局終了時に盤面全体における自分の手番 (黒または白) の石が囲っている交点の多さ (地、スコア) で勝

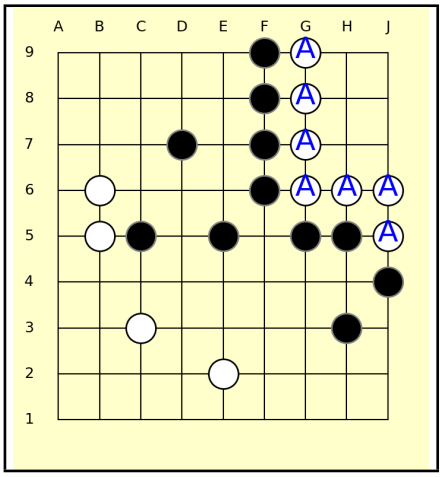


図 1: 詰碁問題の例: 右上の「A」が描かれた白石の群の生死は局所問題であるが、ゲームの勝敗に大きく関わる

[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
[a)] murakami315@g.ecc.u-tokyo.ac.jp
[b)] kaneko@acm.org

敗を決める競技である[*1]。一方、詰碁（死活問題）は主にある局所的な条件設定において相手の石を取ることを目的とする問題であり、スコアを競う囲碁というゲームそのものとはやや異なる目標を持つ。例えば、図 1 は通常の囲碁の一場面であるが、「A」が付与された白石の群を取ることを目指す「詰碁」としても捉えることが可能である。この例のように、対局中の局面に含まれる詰碁は部分問題であると同時に囲碁の勝敗にも大きく影響しうるため、ゲームに強い（勝利できる）エージェントは詰碁を解決する能力を持つことが望ましい。

注目範囲における勝敗を評価するための指標である範囲占有率 (region occupancy) を独自に定義し、範囲占有率をラベルとした Gumbel AlphaZero 型の学習サイクルによってモデルを訓練した。結果として、注目範囲を指定して訓練されたエージェントは従来の勝敗のみを報酬としたエージェントに対し、注目範囲内で勢力を築く能力が高く、提案手法で訓練したエージェントは実行時に指定した region に対応した着手を行った。

2. 先行研究

AlphaGo [7], AlphaZero [6], KataGo [9] など深層強化学習に基づいた多くのエージェントはゲームの勝利を報酬とし, 前節で述べたような高い成果を発揮している。それに対して、詰碁の解決を目的とした探索アルゴリズム (詰碁 solver) は XUanXuanGo[*2], パンダ先生などのサービスが利用可能であり、[*3] 詰将棋の解決に用いられる df-pn を適用した岸本らの研究 [4] やその発展 [5], [10] 等が存在しており、高度な問題も題材とされている。その一方で、df-pn は、通常の対局とは完全に別のアルゴリズムである。

本研究は主に AlphaZero, Gumbel AlphaZero の学習を踏襲している。AlphaZero, Gumbel AlphaZero におけるモデル構造はゲームの局面 S を入力として、S に対して良いと思われる着手を確率分布で表す方策 P_S と S に対する勝率の予測値を表す評価値 V_S の 2 つの出力を持つ。そうした構造を持つモデルは図 2 に示すように AlphaZero, Gumbel AlphaZero はサイクル型の学習プロセスによって訓練される。各サイクルでは訓練されたモデルによる自己対戦データ（棋譜）を蓄積し、自己対戦データを用いてさらにモデルを訓練するというサイクルでによって訓練される。本稿では各時点のモデルをそのサイクルを実行した回数で第 i 世代のモデルと呼称する。モデル訓練時には最新の n 世代分の自己対戦データからランダムにサイズ k のデータを抽出して次世代のモデルの訓練データとし、第 i 世代までの自己対戦データで訓練されたモデルを第 $i+1$ 世代のモデルと呼称する。

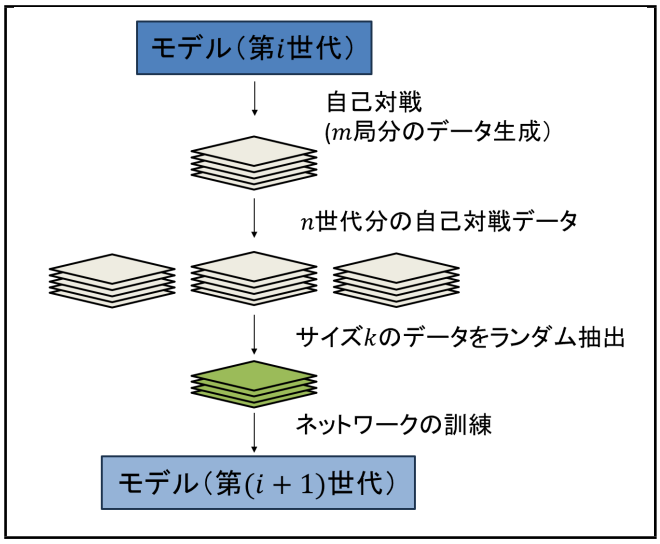


図 2: 学習サイクル: 第 i 世代目のモデルが m 局分の自己対戦を行い, 最新の n 世代分の自己対戦データからランダムに抽出されたサイズ k のデータで訓練されたモデルを第 $i+1$ 世代目のモデルとする. 提案手法においては自己対戦の部分で対戦開始時に注目範囲が事前に決めた候補のうちの 1 つに指定される.

また、KataGo は様々なコミでの対局を総合して訓練を行う。この点は次に述べる本研究の手法で様々な注目範囲の棋譜で対局した棋譜を総合して訓練する点と類似性がある。

3. 提案

本稿では AlphaZero 型の深層強化学習を自然に拡張し、対局の勝利だけでなく実践詰碁（囲碁の対戦を行う中で自然に発生しうる詰碁の意）にも対応したエージェントの学習手法を提案する。対局と詰碁を同時に行うために、ネットワークの入力を拡張して盤面の注目する範囲 (region) を指示する。これが全部 0 なら通常の対局を、そうでなければ 1 の交点が注目範囲とする。範囲 (region) を適宜変更することで一つのモデルで任意の部分に注目できる。注目範囲に応じてネットワークが適切な着手や価値を出力できるよう、範囲占有率 (region occupancy) を定義し、それに基づいてポリシーと価値の訓練を拡張する。拡張部分以外は AlphaZero または Gumbel AlphaZero [1] 型の自己対戦とモデル訓練のサイクルを継承する。棋譜を生成する自己対戦においては、注目範囲の無い通常の対局と、いくつかの範囲を指定した対局を混合して行う。

これにより、通常の対局での強さと多様な注目範囲への対応力の両立を目指す。以下、拡張部分について詳細を説明する。

[*1] 一方、コンピュータプログラムの作りやすさから、石の数を数えるルールが採用されることも多く、本稿もそれに倣っている。両者のルールは概ね同じ勝敗となるので、本稿ではその違いには立ち入らない。

[*2] https://lifein19x19.com/viewtopic.php?f=9&t=16411

[*3] https://www.nikkei.com/article/DGXMZO04612940Y6A700C1000000/

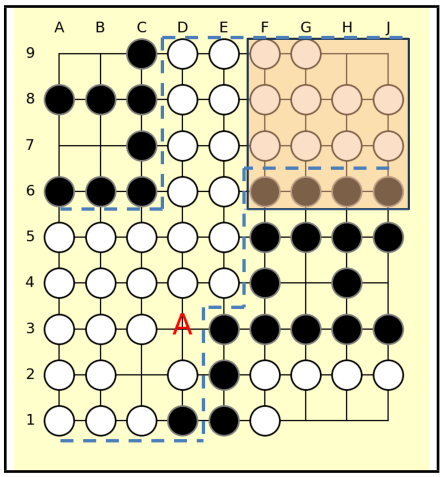


図 3: region 外の着手が必要な局面: 上の図において右上の白石の群は右上から左下まで繋がっており、右上の白を殺すために、黒は左下 (A) に着手する必要がある

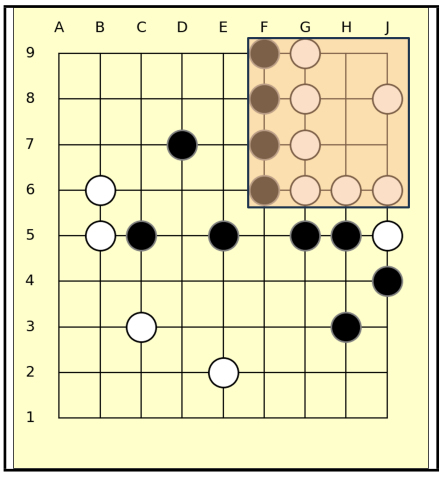


図 4: 範囲占有率: 上の図において region を右上 1/4 とした場合, 範囲内に黒石は 4 子, 白石は 7 子存在するため, 範囲占有率は黒番の場合 $\frac{4}{4+7} \approx 0.36$, 白番の場合 $\frac{7}{4+7} \approx 0.64$

3.1 モデルの概要

AlphaZero 型のモデルがゲームの局面 S を入力として、S に対して良いと思われる着手を確率分布で表す方策 P_S と S に対する勝率の予測値を表す評価値 V_S を出力する一方で、提案モデルはゲームの局面 S に注目範囲 (region) を追加した S_r を入力して S_r に対する region を考慮した場合に良いと思われる着手を確率分布で表す範囲方策 P_{S_r} と S_r に対する region での勝率の予測値を表す範囲評価値 V_{S_r} を出力する。region における勝率は範囲占有率によって判断され、概ね region 内における石の生き死にに相当する。そのため、提案モデルは基本的に region 内で自分の石を生かし、相手の石を殺すような着手を選択することが期待される。ただし、図 3 のように石が region の外側まで繋がる群を形成している場合には該当する群の生き死にに干渉するために region の外での着手を選ぶことも期待される。

また、提案モデルは region を「無し」(null) とした場合には通常の AlphaZero モデルと同様に、通常の囲碁としての対戦を行う。

AlphaZero, Gumbel AlphaZero の学習サイクルでは訓練されたモデルによる自己対戦データ（棋譜）を次のモデルの訓練データとする。提案モデルでは訓練されたモデルを用いて対局開始時に与えられた region に基づく棋譜を次のモデルの訓練データとする。このとき、一局を通して両プレイヤーに同一の region が与えられ、両プレイヤーが囲碁の勝敗ではなく同一の region における勝利を目指してプレイした場合の棋譜が得られる。また、モデルの訓練には範囲方策 P_{S_r}, 範囲評価値 V_{S_r} に対応するための loss を設計し、使用した。このとき、範囲評価値 V_{S_r} の loss は 3.3 に定義する範囲占有率をラベルとした最小二乗誤差である。

3.2 region の設定

本論文における region は右上 1/4(ru), 左上 1/4(lu), 右下 1/4(rd), 左下 1/4(ld) の 4 種類として手動で設定した。実験では 9 路盤を使用し、各 region の大きさを囲碁がゲームとして成立する最低限の広さである 4×4 とするためこの 4 種類とした。19 路盤において各 region の大きさを 4×4 とする場合の regon は $3 \times 3 = 9$ つとなる。ただし、本論文における region の設定はあくまでも一例であり、右半分、左半分、中央など他の設定も可能である。

3.3 範囲占有率の設定

region における勝敗を測る指標として範囲占有率 (region occupancy) を定義した。region occupancy(R_o) はゲームの i 手目 ($i \geq 0$) の盤面において region 内に存在する黒石の数を b_i, 白石の数を w_i とした際に式 (1) で表される。

$$R_o = \begin{cases} \frac{b_i}{b_i+w_i} & \text{if black to move,} \\ \frac{w_i}{b_i+w_i} & \text{if white to move.} \end{cases} \quad (1)$$

ただし、モデル訓練時にはラベルを $(2R_o - 0.5)$ とし、値域を従来の AlphaZero モデルと同様の [-1, 1] に調整した。

3.4 モデルの詳細

変更されたモデルの構造を図 5 に示す。入力には従来の AlphaZero モデルに対する入力として使用される現在から過去 h 手分の盤面上の石配置 (手番別)$2 \times h = 16$ 面に手番を示す 1 面を合わせた計 $2h + 1$ 面に加え、region 内を 1 とし region 外を 0 としたチャネル ($H \times W, H, W$ はそれぞれ盤面の行数および列数) を追加した計 $2h + 2$ 面 ($(2h + 2) \times H \times W$) である。

出力は方策 P, 評価値 V の二つから範囲方策 P_r, 評価値 V, 範囲評価値 V_r の三つに変更した。3.1 節で述べたように評価値 V は従来の AlphaZero モデルと同様にゲームの勝敗の予測値であり、範囲評価値 V_r はゲームの最終局面における範囲占有率の予測値である。コミによる影響が範

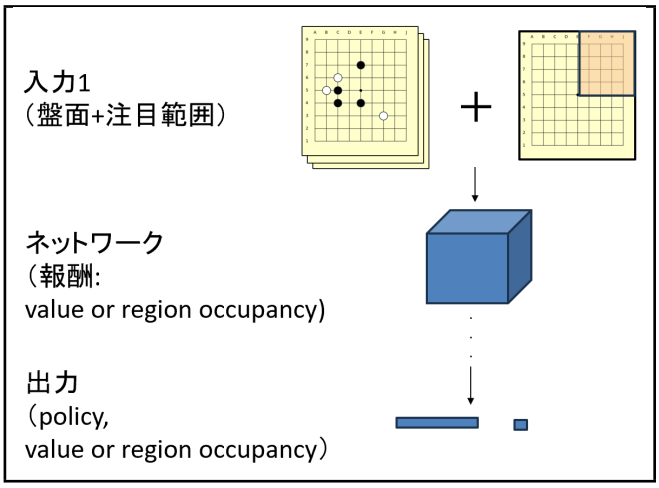


図 5: 提案手法におけるモデル構造: 入力に region(9×9 の追加, 出力に範囲占有率を推測する head の追加

囲評価値に及ぶのを避けるため、V を予測する head と V_r を予測する head は別とした。

対局や訓練において、region を表すチャネルの値を全て 0 にした場合には、region は「無し」(null) と判断され、出力は V が使用され、region が存在する場合は出力として V_r が使用される。

4. 評価

提案手法を元にエージェントの訓練を行い、エージェント同士による対戦成績と範囲を指定した場合の占有率ならびに詰碁の解決性能を観察した。

4.1 提案手法モデルの訓練

提案手法の有効性を評価するために、学習条件を部分的に変更した 5 種類のエージェントを評価した．先に共通部分を述べたのちに、各モデルを定義する．共通の学習部分は、エージェントのモデル構造とエージェント A, B, C の訓練に使用される既存の棋譜データ、エージェント A, C, D, E の訓練で行われる自己対戦の設定の 3 つである。

モデルの構造は、入力チャネル数は現在から過去 8 手分の盤面上の石配置（手番別）を示す $2 \times 8 = 16$ 面に手番を示す 1 面と region を示す 1 面を合わせた計 18 面 $(18 \times 9 \times 9)$ であり AlphaZero を模したモデルとした。

残差ブロックは Gumbel MuZero に倣い、ボトルネック構造を導入し、チャネル数は 128 と 64、ブロック数は 5 の Gumbel MuZero における 9 路盤の実験の縮小版を利用した[*4]。

棋譜データについて、エージェント A, B, C の訓練において自己対戦以外の棋譜データを使用する部分では既存の約 10 万局の棋譜 [*5]を使用した。

次に, エージェント A, C, D, E における自己対戦時の諸条件について述べる。エージェント A では第 1 世代の自己対戦を行う前に既存の棋譜に対して範囲占有率を右上 1/4, 右下 1/4, 左上 1/4, 左下 1/4 と region 無し（null,region を無しにした場合、ラベルは通常の囲碁の勝敗となる）の五種類にラベル付けしたデータからランダムに抽出された 100 万局面分のデータを使用して訓練した。エージェント C では第 1 世代の自己対戦を行う前に既存の棋譜データを、範囲占有率によるラベル付けをせず、従来の AlphaZero 型学習サイクルと同様に、ゲームの勝敗をラベルとして使用し訓練を行った。エージェントの各世代は 1 万局の自己対戦を行い、最新の 100 世代分の対戦データからランダムに抽出された 100 万局面分のデータを次の世代の訓練データとした。また、自己対戦は候補手 8 の Gumbel AlphaZero のアルゴリズムによって着手を決定した。自己対戦と既存の棋譜データの結果の region 無しのゲームの勝敗はいずれもコミを 7 目、Tromp Taylor ルール[*6]で集計を行った。

エージェント A　region は 9 路盤の右上 1/4, 右下 1/4, 左上 1/4, 左下 1/4 と region 無し（null,region を無しにした場合、ラベルは通常の囲碁の勝敗となる）の 5 種類として提案手法による学習サイクルによって訓練を行った。第 1 世代の自己対戦を行う前に既存の棋譜で事前に訓練を行った。

エージェント B　既存の棋譜（約 10 万局分）の着手と勝敗をラベルとして通常の教師あり学習を 10 エポック分行った。このモデルでは強化学習や注目範囲を指定した訓練は行わない。

エージェント C　region を region 無 (null) のみとして提案手法による学習サイクルによって訓練を行った。第 1 世代の自己対戦を行う前に既存の棋譜で事前に訓練を行った。

エージェント D　region は 9 路盤の右上 1/4, 右下 1/4, 左上 1/4, 左下 1/4 と region 無し（null,region を無しにした場合、ラベルは通常の囲碁の勝敗となる）の 5 種類として提案手法による学習サイクルによって訓練を行った。事前の既存の棋譜による訓練は行わなかった。

エージェント E　region を region 無 (null) のみとして提案手法による学習サイクルによって訓練を行った。事前の既存の棋譜による訓練は行わなかった。

4.2 実験 1　対戦性能の確認

通常の囲碁の対局プレイヤーとして提案手法の性能を測定した。提案手法では、注目範囲の学習を行う分だけ通常の勝敗の学習が遅れることが予想され、その影響の程度を確認するためである。具体的には既存の棋譜のみを用いて

[*4] https://github.com/tkaneko/pygo/blob/main/pygo/network.py
[*5] http://www.yss-aya.com/ayaself/ayaself.html#nats2018
[*6] https://tromp.github.io/go.html

表 1: 各エージェントの学習条件

name	A	B	C	D	E
強化学習	有	無	有	有	有
既存の棋譜の使用	有	有	有	無	無
region	5 種類	-	null のみ	5 種類	null のみ

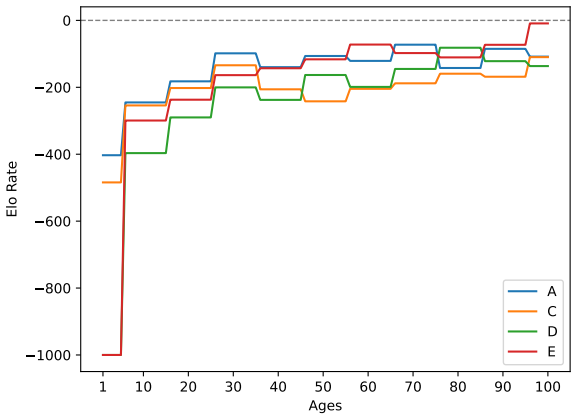


図 6: 通常エージェントとの対戦結果: 横軸はエージェントの世代, 縦軸はエージェント B を Elo rate= 0 とした場合のエージェントの各世代が region を ru とした場合の Elo rate. 第 100 世代時点ではエージェント E の Elo rate が 0 に近くなっており、その他のエージェントの Elo rate は −150 から −100 までの範囲に収まり、訓練時に指定された region の数に関わらず結果に差は見られなかった.

訓練されたエージェント B と提案手法による学習サイクルによって訓練されたエージェント A, C, D, E との対戦を行った。エージェント B を Elo rate= 0 としてエージェント A, C, D, E は各世代ごとに region を「無し」(null) として 1000 局の対戦 (手番は 1 局ごとに交代し、各エージェントが黒番を 500 回、白番を 500 回持つ。エージェント A, B, C の学習に使用した既存の棋譜に倣い、コミは 7.0 目とし、勝敗は Tromp Taylor ルールによって集計) を行った場合の戦績から計算された Elo rate [3] 結果を図 6 に記録した。

結果としては、第 100 世代のエージェント A, C, D は概ね Elo rate は −150 から −100 までの範囲に収まり、大きな差は見られなかった。一方で事前データを用いず、region は「無し」(null) のみを指定した、実質的に Gumbel AlphaZero と同等の構造を持つエージェント E の第 100 世代の Elo rate は約 −9 となり、教師あり学習によって訓練されたエージェント B の性能に近い性能となった。第一世代についてエージェント A, C と D, E の強さに大きな差があった。前二者は既存の棋譜データを第一世代の訓練に用いていることがその理由と考えられる。

4.3 実験 2 region を考慮した対戦性能の確認

提案手法による訓練によって region を考慮した対戦性能が改善されることを示すため region を指定した対戦を行った。エージェント C の第 100 世代で region を「無し」(null) とした場合を Elo rate = 0 として、エージェント A, C, D, E の各世代は右上 1/4 を指定して 1000 局の対戦を行った。(実験 1 と同様に手番は 1 局ごとに交代し、各エージェントが黒番を 500 回、白番を 500 回持つ) Elo rate はゲーム終了時の範囲占有率から算出した。結果としては、右上 1/4 を含む 5 つの region を指定して訓練されたエージェント A, D は世代を経るごとに Elo rate を上昇させていったのに対し、region「無し」(null) のみで訓練されたエージェント C, E の Elo rate は −200 付近に留まった。また、エージェント A, D 間では事前データを使用しなかったエージェント D の方が第 100 世代目時点での Elo rate は高くなった。

次に、提案手法の応用性を観察するため訓練時に使用された region を組み合わせた region を指定して対局を行った。エージェント C の第 100 世代で region を「無し」(null) とした場合を Elo rate = 0 として、エージェント A, C, D, E の各世代は右上 1/4 と左下 1/4 を同時に指定して (region を示すチャネルにおいて盤面の右上 1/4 と左下 1/4 の部分の両方をを region 内を示す 1 とした)1000 局の対戦を行った。(実験 1 と同様に手番は 1 局ごとに交代し、各エージェントが黒番を 500 回、白番を 500 回持つ) 結果として、第 100 世代の Elo rate はエージェント A, D, C, E の順に高くなった。右上 1/4 のみを指定した場合との違いとして既存の棋譜データを使用したエージェント A と既存データを使用しない C の差が開いた事が挙げられる。このことは region が広がるにつれ、ゲーム全体の勝利を目指す場合と同様に盤面全体を意識する必要性があり、そうした観点を持つ既存の棋譜データが学習に有効に働いたためと考えられる。

実験 1 の結果と合わせて考えると、region を指定して訓練されたエージェントはそれ以外のゲームの勝敗を目指すエージェントと比較すると、ゲーム全体の勝利を目指す性能は変わらないもしくはやや劣るものの、region 内において勢力を築く能力は高くなっていることが分かった。

4.4 実験 3 詰碁の解決

図 9 に示す詰碁問題に対して、各世代のエージェントが詰碁を解く（region 内の白石を適切に殺しに行くことができる）か否かを観察した。（詰碁は黒番で 1 つしか正解のない問題を使用した）詰碁の局面に対する各エージェントの範囲評価値は図 10 のようになった。（範囲方策によって正解の着手 (H9) が選ばれなかった場合の範囲評価値は 0 としている) エージェント C,E は世代を経るにつれ、正解の着手を選ぶようになっているものの、範囲評価値は 0.5

付近に留まった。エージェント D は範囲評価値は 0.6 から 0.8 の範囲に存在するものの、正解の着手を選んでいない世代が多い。エージェント A は安定して正解の着手を選択しており、範囲評価値も 0.6 から 0.8 の範囲に存在する。そのため、正解の着手 (H9) を選択することで右上の白石の一群を殺すことができると正しく認識できていると考えられる。

また、図 11 に示すいくつかの詰碁問題ではエージェント B の範囲方策が選ばなかった正解の着手をエージェント A が選択した。このことにより、詰碁問題によってはエージェント A は提案手法によるモデル間だけでなく、教師あり学習による訓練よりも高い性能を示すことが分かった。特に, 図 11 の右側の図において、エージェント B は詰碁問題が存在する右上以外の範囲におけるゲームの勝利を目指す着手を選択していると考えられる。

提案手法によって訓練されたエージェントは入力として与える region を切り替えることで異なる範囲方策を示すため、注目したい部分に特化した着手を得ることが期待される。図 12a の問題に対して、入力として与える region を切り替えながらエージェント A, D が選択する着手を確認した。図 12a の右上、左上、右下、左下の 4 つの白石の群はそれぞれ H9、 B9、H1、 B1 に着手することで殺すことができ、エージェントは指定した region の群を殺す手を選択することが期待される。結果として、図 12b に示すようにエージェント A は左上 1/4(lu) を region とした場合以外は全て右上の白石の一群を殺す手を選択した。その一方で、図 12c に示すようにエージェント D は指定した region に存在する白石の一群を殺す手を適切に選択した。原因としては、囲碁は通常右上への着手から試合が開始されるため、エージェント A の訓練が既存の棋譜に影響を受けている可能性が考えられる。

5. まとめと今後の展望

本稿ではゲームの勝利を報酬とする AlphaZero, Gumbel Alphazero 型のモデルに対して、入力に注目範囲 (region) を追加して訓練を行うことで、AlphaZero 型の強化学習を拡張して部分問題である実践詰碁の解決能力の追加を試みた。df-pn 等の既存の詰碁解決アルゴリズムに対して性能は遠く及ばないが、提案手法の利点としては、調べたい局面に対して本手法を適用する際に実行コストが小さい点や、注目したい範囲の石の生死が注目範囲外の石と関わっているような場合にも対応できる点が挙げられる。

注目範囲内での勝敗を決定するための指標として範囲占有率 (region occupancy) を新しく定義し、それをモデル訓練の際のラベルとして AlphaZero,Gumbel AlphaZero と同様の自己対戦とモデルの訓練による学習サイクルで訓練を行ったところ、通常の対戦性能は従来の勝敗を報酬としたモデルよりやや劣り、region を指定して対戦を行った場合

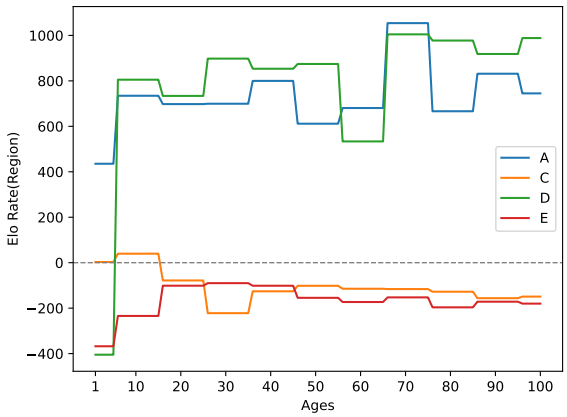


図 7: region を指定した対戦結果: 横軸はエージェントの世代, 縦軸はエージェント C の第 100 世代が region を「無し」(null) として対局した場合を Elo rate= 0 とした場合にエージェントの各世代が右上 1/4 を region として指定し, 対局した場合の範囲占有率から算出された Elo rate. ただし, エージェント C, E については訓練時に右上 1/4 を指定して対局したデータを使用しないことに留意が必要である. 右上 1/4 を含む 5 つの region を指定して訓練されたエージェント A,D は世代を経るごとに Elo rate を上昇させていった.

の性能は高くなった。また、regin を指定した訓練では複数の異なる region に対応するように訓練を行い、対戦時にそれらを組み合わせた場合も従来のモデルよりも高い性能を発揮し、提案手法で訓練したエージェントは実行時に指定した region に対応した着手を行った。

今後の展望としては、df-pn のような長手数の探索を用いた solver との性能差を評価し、近づけてゆく方向や、より適切な注目範囲の選定が挙げられる。本論では通常の AlphaZero、Gumbel AlphaZero の訓練と同様に囲碁の通常の初期状態（盤上に石が無い状態）を初期局面としたが、出村ら [2] のように初期局面や布石段階の局面に変更を加えることによる性能の向上の可能性がある。例えば、自己対戦や訓練データの初期局面を生成 AI 等の手法により生成された詰碁などに変更することで、より高度な問題への対応能力の向上が期待される。また、本論では囲碁の 9 路盤上を 4×4 の正方形として 4 つに分割したが、対称性の無い注目範囲を指定した場合に学習にどのような変化が起きるかも調査したい。

4.3 節の実験で述べたように、注目範囲が広い場合には通常のデータが学習に良い効果をもたらす事が示唆されている。実験に使用したエージェントは各注目範囲で同量のデータを生成し、学習を行ったが注目範囲「無し」の割合を増やすことでより応用力が上がる可能性が考えられ、今後そうした実験を行いたい。

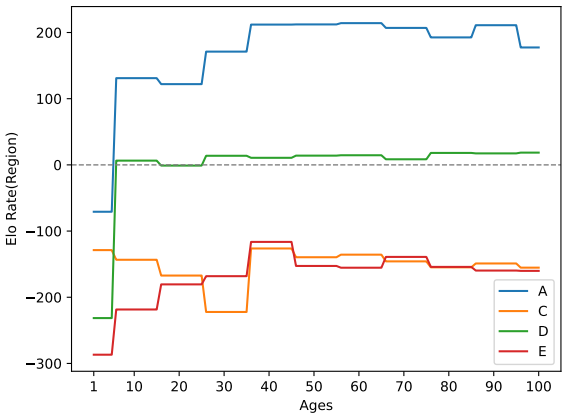


図 8: region を指定した対戦結果: 横軸はエージェントの世代, 縦軸はエージェント C の第 100 世代が region を「無し」(null) として対局した場合を Elo rate= 0 とした場合にエージェントの各世代が「右上 1/4 と左下 1/4」region をとして指定し, 対局した場合の範囲占有率から算出された Elo rate. ただし, エージェント C, E については訓練時に「右上 1/4 と左下 1/4」を指定して対局したデータを使用しないことに留意が必要である.「右上 1/4 と左下 1/4」を含む 5 つの region を指定して訓練されたエージェント A,D は世代を経るごとに Elo rate を上昇させていき, 結果として、第 100 世代の Elo rate はエージェント A, D, C, E の順に高くなった.

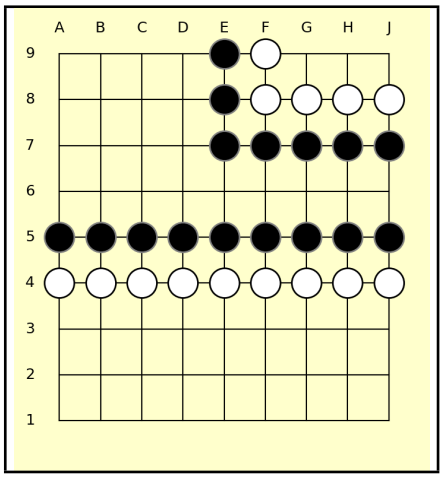


図 9: 詰碁問題の例: 右上の白石の一群は黒が H9 に着手することで殺すことができる. 殺した石は基本的に盤上から取り除かれるため, 範囲占有率は上昇する.

加えて、本論では注目範囲内は 1、注目範囲外は 0 として二値で表現しているものの連続値で表現した場合に注目範囲をやや重視したプレイが可能になると思われるため、これらのアイデアも将来的に挑戦の余地があると考えられる。

謝辞

本論の執筆に際し、匿名の査読者の方、同研究室の出村洋介さん、に多くの助言をいただきました。そのことに関して、この場を借りて深く感謝させていただきます。

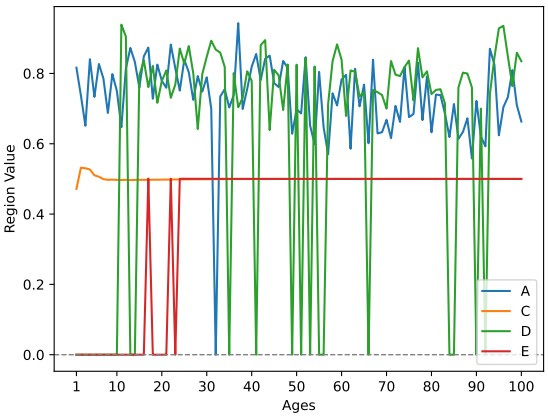


図 10: region を指定した対戦結果: 横軸はエージェントの世代, 縦軸は図 9 に対し,region を右上 1/4 とした場合の範囲評価値. ただし, エージェント C, E については訓練時に「右上 1/4」を指定して対局したデータを使用しないことに留意が必要である. 範囲方策によって正解となる H9 の位置が選ばれていない場合は範囲評価値を 0 とした. エージェント C,E は世代を経るにつれ, 正解の着手を選ぶようになっているものの, 範囲評価値は 0.5 付近に留まった. エージェント D は範囲評価値は 0.6 から 0.8 の範囲に存在するものの, 正解の着手を選んでいない世代が多い. エージェント A は安定して正解の着手を選択しており, 範囲評価値も 0.6 から 0.8 の範囲に存在する.

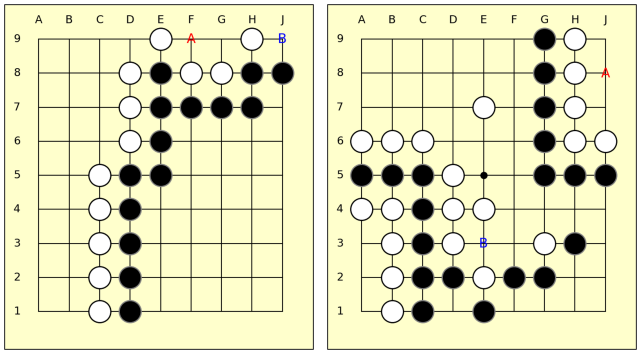


図 11: エージェント A がエージェント B を上回る詰碁の例: 左の図では F9 を選択することで右上の 3 つの白石を取る事ができる. エージェント B が選択した I9 では H9 の白石しか取ることができない. 右の図では I8 の選択により右上の白の一群を殺すことができる. エージェント B は右上の一群に関係なく, ゲームの勝利を目指す着手を選択していると考えられる.

参考文献

[1] Danihelka, I., Guez, A., Schrittwieser, J. and Silver, D.: Policy improvement by planning with Gumbel, *International Conference on Learning Representations*, (online), available from ⟨https://openreview.net/forum?id=bERaNdoegnO⟩ (2022).

[2] Demura, Y. and Kaneko, T.: Initial state diversification for efficient AlphaZero-style training, *ICGA Journal*, (online), DOI: 10.3233/ICG-240255 (2024). in printing.

[3] Elo, A.: *The Rating of Chessplayers: Past and Present*, Ishi Press International (2008).

[4] Kishimoto, A. and Müller, M.: *DF-PN in Go: An Application to the One-Eye Problem*, pp. 125–141 (online), DOI: 10.1007/978-0-387-35706-5_9, Springer US (2004).

[5] Kishimoto, A., Winands, M., Müller, M. and Saito, J.: Game-Tree Search Using Proof Numbers: The First Twenty Years, *ICGA Journal*, Vol. 35, No. 3, pp. 131–156 (2012).

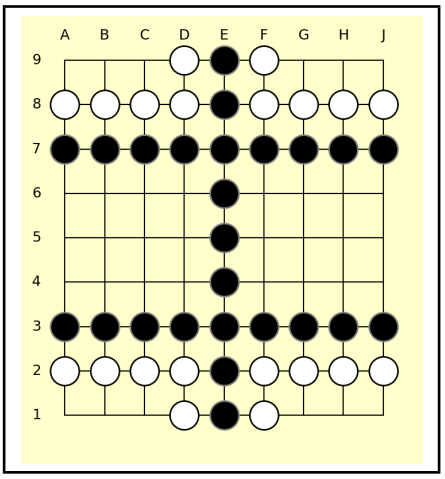
[6] Silver, D. et al.: A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play, *Science*, Vol. 362, No. 6419, pp. 1140–1144 (online), DOI: 10.1126/science.aar6404 (2018).

[7] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T. and Hassabis, D.: Mastering the game of Go with deep neural networks and tree search, *Nature*, Vol. 529, No. 7587, pp. 484–489 (online), available from ⟨http://dx.doi.org/10.1038/nature16961⟩ (2016). Article.

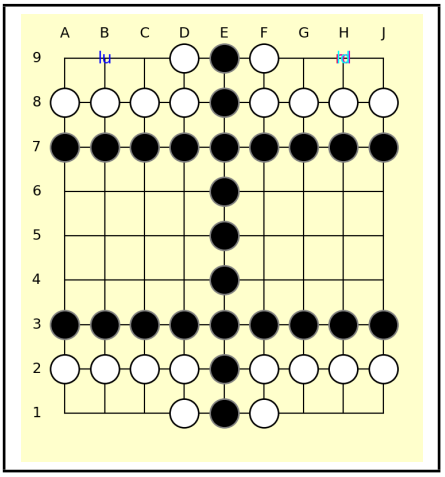
[8] Tian, Y., Ma, J., Gong, Q., Sengupta, S., Chen, Z., Pinkerton, J. and Zitnick, L.: ELF OpenGo: an analysis and open reimplementation of AlphaZero, *Proceedings of the 36th International Conference on Machine Learning* (Chaudhuri, K. and Salakhutdinov, R., eds.), Proceedings of Machine Learning Research, Vol. 97, PMLR, pp. 6244–6253 (online), available from ⟨https://proceedings.mlr.press/v97/tian19a.html⟩ (2019).

[9] Wu, D. J.: Accelerating Self-Play Learning in Go, *CoRR*, Vol. abs/1902.10565 (online), available from ⟨http://arxiv.org/abs/1902.10565⟩ (2019).

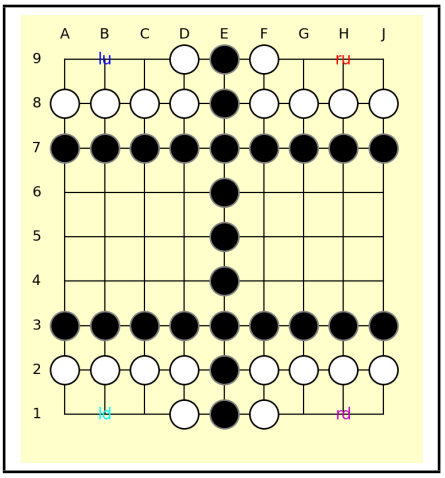
[10] Yoshizoe, K., Kishimoto, A. and Müller, M.: Lambda Depth-First Proof Number Search and Its Application to Go, *IJCAI*, pp. 2404–2409 (2007).



(a) 殺すことができる白石の群が複数ある問題



(b) エージェント A が選択した着手：region を左上 1/4(lu) とした場合以外は該当する左上の白石の群を殺す手を選択した. それ以外の 3 つの region(右上 1/4, 右下 1/4, 左下 1/4) を選択した場合はいずれも右上の群を殺す手を選択している.



(c) エージェント D が選択した着手: 4 つの region である右上 1/4(ru), 左上 1/4(lu), 右下 1/4(rd), 左下 (ld) のそれぞれに該当する白石の群を殺す手を選択している.

図 12: region によるエージェントの出力の変化