# 階層強化学習を用いた説明可能なゲームAI

岩佐拓真[1,a)]　鶴岡慶雅[2]

**概要**：ゲームにおいて相手を打ち負かしたり, 高いスコアを得るような強いゲーム AI の研究はこれまで多く行われているが, 特にプレイヤーを楽しませる目的では, ゲーム AI には単なる強さだけが求められるわけではない. ゲーム AI 分野では今後の発展として, 単なる強さを求めるのではなく, 人間プレイヤーを楽しませたり指導したりする目的を持つゲーム AI の研究が課題の一つとなっている. 本研究ではゲーム AI がどのような戦略に基づいて行動を決定しているのかを説明可能なシステムを実現することを目的とする. 説明可能な AI を実現するための手法は既にいくつか存在するが, それらはゲーム AI に応用するのに困難な部分があるか, 解釈のしやすさに課題が残る. そこで, ゲーム AI を説明するために階層強化学習を用いる手法を提案する. 実験では階層強化学習を用いてブロック崩しの環境で学習を行い, AI のとった戦略を可視化することができた.

# Explainable Game AI using Hierarchical Reinforcement Learning

Takuma Iwasa[1,a)]　Yoshimasa Tsuruoka[2]

***Abstract:*** There have been many studies on game AI that can beat opponents or get high scores in games. However, especially for the purpose of entertaining players, game AI is not only required to be strong. One of the challenges for future development in the field of game AI is to study game AI that has the purpose of entertaining or guiding human players, rather than merely seeking strength. The purpose of this study is to realize a system that can explain what strategy a game AI decides its actions by. There are already several methods to realize explainable AI, but they are difficult to apply to game AI, or they are not easy to interpret. Therefore, we propose a method using hierarchical reinforcement learning to explain game AI. In our experiments, we used hierarchical reinforcement learning in a brick breaking environment and succeeded in visualizing the strategies taken by the AI.

## 1. はじめに

ゲームにおいて相手を打ち負かしたり高いスコアを得るような強いゲーム AI の研究はこれまで多く行われており, 例えば 2016 年にコンピュータ囲碁 AlphaGo [1] が囲碁のプロ棋士に勝利するなど, 一部のゲームでは人間以上の強さを持つ AI の開発が達成された. しかし, 特にプレイヤーを楽しませる目的では, ゲーム AI には単なる強さだけが求められるわけではない. 例えば, 多くのゲームにおいて敵キャラクターはプレイヤーによって倒されるために存在している. 箭本 [2] によればゲームの相手を務める AI は，わざと隙を作ってプレイヤーの攻撃を受けたり，プレイヤーが攻略しやすいパターンで動いたりしたうえで，勇猛だったり臆病だったりといった個性まで表現しなければならない. 逆に, 敵キャラクターの AI が常に最適な行動をとる場合, プレイヤーは勝つことができずゲームプレイを楽しめない. また勝ち負けの面以外でも, 池田 [3] によって述べられているように, 人間ではまず着手しないような手が AI から出てくることはしばしばあり, それは楽しい意外性になりうるが，そうでない場合も多い. また AI 側が手加減をする場合は, 明らかに手抜きだとわかってしまい楽しさを減じる結果になってしまうことも多い. このように人間的でない不自然な行動をとる AI がプレイヤーに違和感を与え,

[1] 東京大学工学部電気電子工学科
Department of Electrical and Electronic Engineering, The University of Tokyo
[2] 東京大学大学院情報理工学系研究科電子情報学専攻
Graduate School of Information Science and Technology, The University of Tokyo
[a)] iwasa@logos.t.u-tokyo.ac.jp

©2022 Information Processing Society of Japan

ゲーム体験に悪影響を及ぼすこともある. 以上の, 人間プレイヤーを楽しませる目的を持つゲーム AI のように, ゲーム AI 分野では今後の発展として単なる強さ以外を目的とする研究が課題の一つとなっている. ゲームについて, 人間に分かりやすいような解説を行う AI もこの一例である. これまでも, 将棋のルールを知らない, あるいは経験が少ない観戦者にも分かるように, プロ棋士の対局の局面評価と指し手の予想を提供して解説するシステムの研究 [4] や, 囲碁の局面の解説文を生成するための適切な用語を選択する AI の研究 [5] などが行われている.

本研究ではゲーム AI がどのような戦略に基づいて行動を決定しているのかを説明可能なシステムを実現することを目的とする. AI の判断を説明するための手法は既にいくつか存在するが, それらはゲーム AI に応用するのに困難な部分があるか, 解釈のしやすさに課題が残る. そこで, ゲーム AI を説明するための新しい手法を提案し, その有効性を検証する.

## 2. 関連研究

### 2.1 説明可能な AI

説明可能な AI は, 機械学習モデルが入力を受けてから出力が出るまでのプロセスを人間にも理解できるようにする技術である. 谷岡 [7] によれば, 人工知能のシステムが採用するほとんどの機械学習アルゴリズムでは, 学習結果として得られるモデルについて, 決定木やランダムフォレストのような説明変数の寄与度を算出可能なアルゴリズムを用いないと, 解釈不可能な場合が多い. このため構築された予測モデルはブラックボックスとなり, なぜそのような予測が得られるかについて説明できないという事態に陥る. 医療, 金融サービス, 司法などの分野にそのような AI を用いる場合, AI による判断の根拠を説明できなければ利用する人間に信頼感や納得感を与えることは難しい. ゲーム AI の分野においても, 松原 [6] はディープラーニングが持つ構造的な問題点として, どうやって解答にたどり着いたかの筋道を示さないことを挙げており, AlphaGo は非常にいい手を打っているものの, 人間のプロ棋士のように指し手の意味を解説することができないとしている. 説明可能な AI によってこのような課題を解決することができる. 説明可能な AI を実現するための手法のうち, いくつかを以下に示す.

#### 2.1.1 大域的な説明

大域的な説明では, 深層学習モデルなどの複雑なモデルを可読性の高いモデル, 例えば単一の決定木やルールモデルで近似的に表現することでモデルの説明とする. 決定木は木構造でデータを上から各クラスに分類していく手法で, 予測の道筋が分かるため理解が用意になる. 例えば BornAgainTrees [8] では, 最初に予測精度の高いニューラルネットなどのブラックボックスモデルを構築し, その後学習したモデルを使い擬似訓練データを生成する. この時入力データである特徴量からランダムにサンプリングを行い, 学習モデルで予測させ, サンプリングデータと学習結果を疑似訓練データとする. その後生成した擬似訓練データを利用して解釈性の高い決定木モデルを構築し, 2 つのモデルから予測精度と解釈性を実現する.

#### 2.1.2 局所的な説明

局所的な説明では, ある入力からモデルが予測を出したときに, その根拠を説明する. その方法の一つは, データの各特徴量が出力にどの程度の影響を与えているかを示すことである. これにより影響の大きい特徴量を判断の根拠として説明できる. また, 誤った判断を行った際も, 判断の根拠となった特徴量を可視化することでその理由を説明できる.

LIME [9] は, あるデータの近傍に限れば, 複雑な AI モデルであっても単純な線形モデルで近似できると仮定している. 線形モデルを用いることで係数の比較から各特徴量の重要度を測ることができる. LIME の動作原理は大きく以下の 3 ステップに分けることができる.

1. 説明対象のデータをランダムに摂動させて, 近傍データを生成する.
2. 近傍データに対する説明対象 AI モデルの予測結果を取得する.
3. 近傍データと 2. の結果を組み合わせたデータを用いて, 解釈可能な線形モデルを獲得する.

SHapley Additive exPlanations (SHAP) [10] は, 同様にモデルの予測結果に対する特徴量の寄与を求めるための手法であり,Shapley 値と呼ばれる考え方に基づいている.Shapley 値は元々協力ゲーム理論と呼ばれる分野で提案された分配手法の一つである. 協力ゲーム理論では, 複数のプレイヤーが協力してクリアすることで報酬が得られるようなゲームにおいて, 各プレイヤーの貢献度に応じて報酬をいかに公平に分配するかを求めることが主なタスクとなっている. 機械学習ではいろいろな特徴量が組み合わさって予測値が算出されているので, ゲーム＝ 1 行のデータ, プレイヤー＝特徴量, 報酬＝予測値と読み換えることで, 機械学習においても Shapley 値の考え方を利用することができ, SHAP では「それぞれの特徴量が予測値にどれだけ影響を及ぼしているか」を, 各特徴量が予測値を平均よりどのくらい上昇または下降させたかによって測る.

特徴量以外の根拠として, Koh ら [11] により, 影響関数を用いて, 個々の学習データの有無や摂動が予測結果に与える影響を定式化する手法が提案されている. すべての学習データを学習したネットワークと, ある学習データを学習しなかったネットワークにテストデータを入力したとき

©2022 Information Processing Society of Japan

の損失を比較することで, その学習データが予測結果に与える影響を調べることができる. また, ある学習データが変化したときのネットワークのパラメータや損失の変化量も計算することができる. この手法を用いることで, ネットワークの挙動の分析だけでなく, データセットの操作や学習データの誤りの検出をすることもできる.

#### 2.1.3 説明可能なモデルの設計

大域的な説明, 局所的な説明は, ブラックボックスモデルを対象にそこから説明を生成することを目的としている. これに対し, このアプローチでは最初から可読性の高い解釈可能なモデルを作ることを目的とする. Certifiably Optimal Rule List (CORELS) [12] という手法ではルールリストという決定木の亜種を学習する. 扱うデータとしてカテゴリカルデータを想定し, カテゴリカルな特徴空間でルールリストを構築するために, 組み合わせ最適問題を各種探索の枝刈りを用いて高速化を行い, 最終的に解釈性の高いモデルを構築していく. Bien ら [13] は分類問題の各カテゴリを代表する訓練データを検出する方法を提案している.

#### 2.1.4 深層学習モデルの説明

深層学習モデルの説明は, 特に画像認識の分野で数多く研究されている. 基本的には, モデルが画像内のどの部分を認識しているかを特定してハイライトすることで説明とする. 出力ラベルに対する入力画像の勾配を計算し, ある特定の入力画素の微小変化が出力ラベルを大きく変化させる場合に, 対象画素を認識対象であるとしてハイライトする. ただし, 単純に勾配を計算するとノイズの多いハイライトが生成されるので鮮明化させるために, GuidedBackprop [14] や IntegratedGrad [15] のような手法が提案されている. Greydanus らの研究 [16] では, ブロック崩しゲームである Breakout など複数の Atari のゲーム環境において, ゲーム画面の一部分に摂動を加えた画像を用いて, 画面のどの部分が AI の方策に影響を与えているかを可視化し, AI の戦略を解釈できるようにした.

### 2.2 強化学習

強化学習は機械学習の手法の一つである. 機械学習のうち, 教師あり学習では入力と正しい出力が紐づいた学習データを与え, ある入力を与えたときに正しい出力を返すアルゴリズムを構築する. 教師なし学習では入力データのみが与えられ, データの中に内在するパターンなどを抽出して, データのグループを作るクラスタリングや次元削減を行う. それらと異なり, 強化学習では環境が与えられ, その中で設定された報酬を最大化するための行動を学習する. 強化学習の様子を図 1 に示す. エージェントは環境の中で行動を行い, 環境はその行動によって変化し, エージェントに環境の状態と行動に対する報酬を与える. これを繰り返し, 得られる累積報酬を最大化するような行動の選び方 (方策) を学習する. 強化学習では, 一般的にマルコフ決定過程 (Markov Decision Process, MDP) の考え方が用いられる. マルコフ決定過程では, 次の状態への遷移が今の状態と行動のみ依存し, 以前の状態や行動に関係なく行われる. 報酬和とは, 各タイムステップごとの報酬を加算したものをいい, 現在から未来にかけて得られる報酬を全て同じスケールで加算して得られる累積報酬和と, 割引率によってタイムステップごとの報酬の値を割引きながら加算していく割引報酬和が存在する. 行動価値関数とは, マルコフ決定過程においてある状態である行動を取ったときに期待される累積報酬を示したものである. Q 学習 [18] という手法では, 行動価値関数を更新することで学習を進める. Deep Q-Network (DQN) [19] は行動価値関数をニューラルネットワークを用いて求める手法で, Breakout など Atari の複数のゲーム環境において人間を超えるスコアを得ている. しかし, Montezuma's Revenge ではほとんどスコアを獲得できていない. Montezuma's Revenge は 1 エピソードあたりに必要な行動の数が多く, 更に複雑な手順を踏まないと報酬が獲得できず, プレイヤーの死亡によって高い頻度でエピソードが途中で終わるため, エージェントが環境から報酬を得る機会が少ないという特徴がある. 学習が起こるにはランダムな動きの中で複雑な手順を満たして点を得る必要があり, DQN による学習が進まなかったと考えられる.

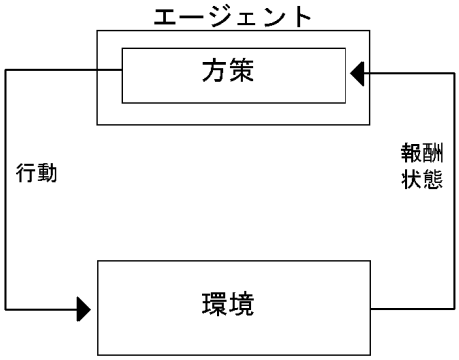


**図 1** 強化学習におけるエージェントと環境

#### 2.2.1 階層強化学習

階層強化学習は, Montezuma's Revenge のように目標を達成するまでに必要な行動が多い, 報酬を得る機会が少ないなどの理由で学習が難しい問題に対して, 階層的な構造を導入することで, 効果的に学習を進める手法である [17]. 階層強化学習の様子を図 2 に示す. 階層構造は上位の方策と下位の方策からなり, 上位の方策は, 達成するべき課題を複数のサブゴールに分割し, 次に到達するべきサブゴールを

©2022 Information Processing Society of Japan

選択する. 下位の方策は, サブゴールを達成するためにエージェントがとるべき行動を選択する. 上位方策の学習では, サブゴールの達成がタスク全体における達成度にどれだけ影響を与えたかに応じて報酬が与えられ, 下位方策の学習ではエージェントの行動によってサブゴールがどれだけ達成できたかによって報酬が与えられる. これにより, 課題を解決するのに膨大な数の行動が必要になる場合でも, 少ない数のサブゴールを検索する課題として考えることや, 複数のタスクの一部をサブルーチンとして共有することができ, 学習を効率よく行える. Le ら [20] は上位方策の学習を模倣学習で, 下位方策の学習を強化学習で行う階層的強化学習の手法 Hierarchically Guided DAgger/Q-learning (hg-DAgger/Q) を提唱した. hg-DAgger/Q は DQN がスコアを全く獲得することができなかった Montezma's Revenge において好成績を収めている.

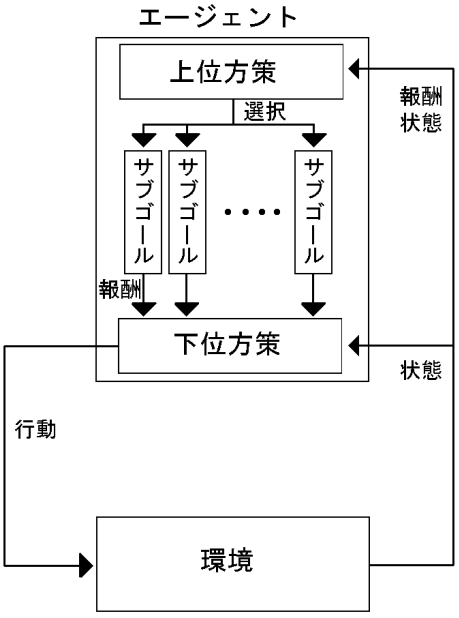


**図 2** 階層強化学習におけるエージェントと環境

高田ら [21] によれば, 人間の計画は典型的には階層的である. 例えば旅行計画は,「目的地に移動」,「目的地の観光」,「自宅に移動」などの目標からなり, さらに「目的地に移動」は,「最寄り駅に移動」,「切符の購入」などより特定された副目標からなる階層的な木構造をしている. 一般的に, 木は部分木に分割して独立に扱うことができる. そこで, 計画全体を記述した木を一括して考慮することなく, 副目標を表す部分木のみを考慮の対象とすることができる. 例えば, ほかのすべての副目標を考慮することなく,「最寄り駅に移動」のみを考慮の対象として, その手段を選ぶことができる. 一杉ら [22] はこの, 人間が再帰的にサブゴールを設定するという振る舞いにヒントを得て, 階層強化学習のアーキテクチャとして RGoal アーキテクチャを提案した. 取りうる状態のうちいくつかをランドマークとして選び, 全体のゴールやサブゴールはそのうちから選ばれるようにし, 再帰的にサブゴールを設定しながらおおもとのゴールを目指す. アルゴリズムは, 現在のサブゴールやサブゴールの変更を取り入れた拡張状態行動空間上の MDP を解く形で定式化される. 行動価値関数は, サブゴールの前後で価値関数を分解することにより複数のタスク間で共有可能になり, マルチタスク環境での学習を効率化する.「思考モード」と呼ばれるモードでは, 一つ一つの行動をシミュレートせずに, 隣接したサブゴール間の移動のみを考えることで近似解を得ることが出来る.「思考モード」における振る舞いは一種のモデルベース強化学習であり, 学習済みのタスクを組み合わせることで, 一度も経験したことのないタスクを少ない試行錯誤で, 場合によってはゼロショットで解くことができる. アルゴリズムはスタックを用いず, フラットなテーブルとシンプルな操作の繰り返しで実現される.

## 3. 提案手法

本研究では図 2 に示した階層構造の方策を用いてゲームの環境で学習を行い, 上位の方策が選択したサブゴールを可視化することで AI のとる戦略を説明する手法を提案する. 2.1 節で示したように説明可能な AI を実現する手法は既にいくつか存在するが, 複雑なモデルを可読性の高いモデルで近似的に表現する大域的な説明では, 一度モデルを作ったあとそれを別のモデルで近似する必要があり, 手間がかかる. そのため, 最初から説明が可能なモデルを設計するのが望ましいと考えられる. 本研究では, 説明が容易なモデルとして階層強化学習の構造を用いる. 階層強化学習では, 上位の方策は課題を達成するために次に目指すサブゴールを選択し, 下位の方策はサブゴールを達成するためにエージェントの行動を選択する. このため, エージェントの行動という出力が, サブゴールを達成するための行動として説明できる. また, 課題をいくつかのサブゴールに分割する方法は人間が階層的に計画を立てるプロセスと類似しており, 人間にとって理解しやすいと考えられる. よってサブゴールを可視化することで, AI の行動の目的を説明できると考えられる. 図 3 に示すように, LIME や SHAP などの既存の局所的な説明では特徴量や学習データ, Greydanus らの研究 [16] では画像内のハイライトされた一部分などといった入力データの一部を判断の根拠として示しているのに対し, 本手法ではエージェントの行動を決定するプロセスの一部であるサブゴールを可視化することで, より直接的にエージェントの行動の目的を示し, 戦略を説明することができると考えられる.

©2022 Information Processing Society of Japan

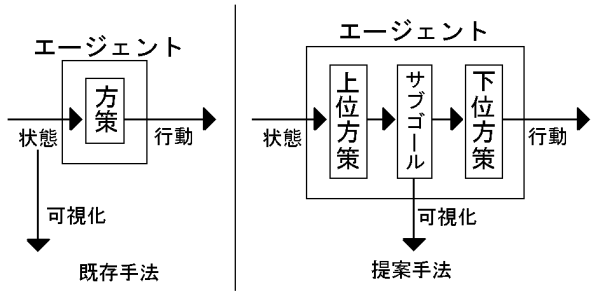


**図 3** 提案手法と既存手法の比較

## 4. 実験

### 4.1 環境

実験では図 4 のようなオリジナルのブロック崩しの環境を用いた. 図において緑色で示されるボールは初期状態では下向きの速度を持っており, 1 ステップごとに速度の値だけ移動し, 画面上端や左右の壁やブロックにぶつかると逆方向に跳ね返り, ブロックにぶつかった場合はそのブロックが壊れる. エージェントは図の下部に青色で示されるバーを動かし, 上部に赤色で示される 6 × 18 個のブロックを壊すことを目的とし, 全て壊すことでゲームが終了する. エージェントの行動はバーを左右へ移動させるか, もしくは動かさないかの 3 通りである. 観測する状態として 6 × 18 個のブロックが壊されているかどうか, ボールの $x$, $y$ 座標, $x$, $y$ 方向の速度, バーの $x$ 座標の情報を用いる. ボールがバーにぶつかった場合, 衝突前の角度によらず, バーのどの位置にボールが衝突したかによってボールの反射する角度が変わる. これによりプレイヤーはある程度狙った方向へボールを跳ね返すことができる. これはゲームの進行のうち, エージェントの行動に左右される部分を大きくすることで, 学習によってブロックを壊すスピードに差異が出やすくなることを狙ったものである.

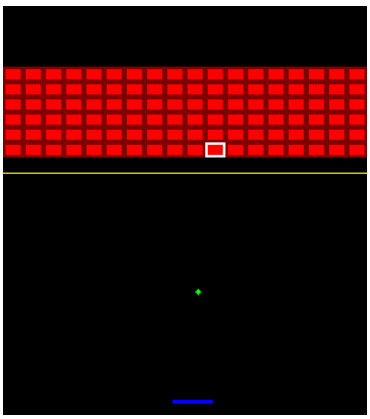

**図 4** 実験に用いた環境

### 4.2 エージェント

実験に用いる階層的な方策を図 5 に示す. サブゴールを決定する上位方策では, 下位方策によってボールを跳ね返して狙う位置をブロックと同じ 6 × 18 通りの座標から 1 つ選択する. なお, 既にブロックが壊されている座標も選択することができる. 上位方策の決定を可視化するため, 選択された位置は白い枠で表示される. 上位方策によるターゲットの選択はボールが図 4 の黄色の線の上部から下に移動するときに行われ, バーがボールを跳ね返し, ボールがブロックや上の壁とぶつかって黄色の線より下の領域に戻ってくるとき, 再び上位方策によって次の目標を決定する. 実験では上位方策として PPO [23] のモデルを用い, 上位方策の学習を行った. 短時間で多くのブロックを壊すことを目的とするため, バーで跳ね返ってから戻ってくるまでに壊したブロックの数を上位方策の報酬とした. ただし, 長時間ブロックを壊さずにいることを避けるため, ブロックを 1 つも壊さなかった場合は報酬を-1 とした. エージェントの行動を決定する下位方策ではルールベースのアルゴリズムを用いる. ボールが黄色の線より上にある間や, ボールが上向きに移動している間はボールがどのように移動しても追いつけるように, バーが常に画面の中央とボールの位置の中間にあるように移動し, ボールが黄色い線の下に移動しターゲットが決定されたあとは, ボールの座標と速度から軌道を計算し, 狙うべき座標から跳ね返す角度を求めて, バーのいるべき位置を判断し, その位置を目指してバーを移動させる.

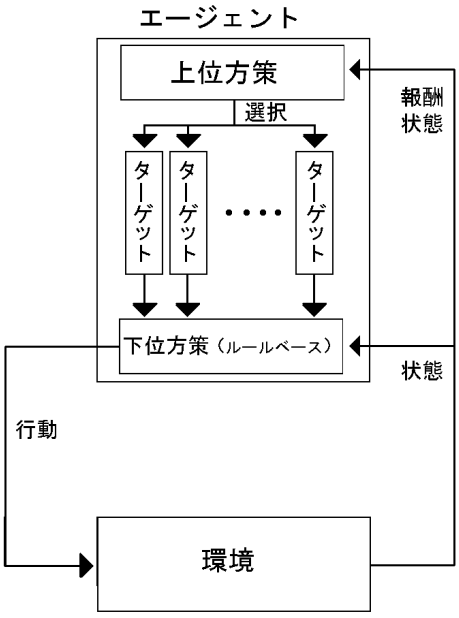


**図 5** 実験におけるエージェントと環境

### 4.3 結果

上位方策によるターゲットの選択で数えて 1,000,000 回

©2022 Information Processing Society of Japan

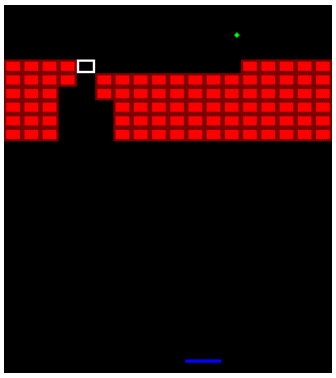

**図 6** ゲーム序盤

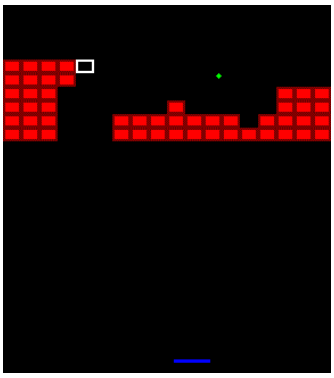

**図 7** ゲーム中盤

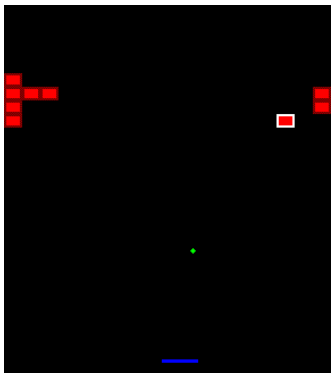

**図 8** ゲーム終盤

ぶんの学習を行った. 学習中の 1 エピソードにかかるステップ数を図 9 に示す.1 エピソードは初期状態からゲームが終了するまでを意味する. 下位方策がルールベースのため, ボールは画面外に落ちることはなく, いつかは必ずすべてのブロックが壊されることになる. そのため 1 エピソードのステップ数はボールを全て壊すまでにかかる時間を表す. 学習が進むにつれステップ数は減少するが, 25,000 ステップ付近に限界を見ることができ, 十分に学習が進んでいるといえる. 学習したモデルを用いたゲームプレイの様子を図 6-8 に示す. 図 6 から分かるように上位方策によって左側の位置が狙われており, 同じ位置を狙い続けたことでブロックの列に下から上まで通じる穴があいた. その穴を通ることでボールが上の壁とブロックの間の空間に入っていることが分かる. ボールは壁とブロックの間で反射を繰り返し, これによりボールが再びバーの位置に落下する前に複数のブロックを壊すことができる. 図 7 からは, 下側のブロックよりも上側のブロックを多く壊していることが分かり, 一度に多くのブロックを壊せる上側にボールを移動させることを優先している様子が見て取れる. また, 図 8 からは, ブロックのほとんどが壊れて同様の戦法が使えなくなったあとは残っているブロックにボールが当たるようにその近くを狙っていることが分かる. このことから,AI がボールをブロックの上の空間に通して一度に多くのブロックを壊す戦略や, 一つ一つのブロックを狙う戦略を学習していて, それを可視化することができたといえる.

### 4.4 考察

Zahavy ら [24] によれば Breakout の環境において良い戦略はブロックの列に穴を開けてボールを上の空間に通すことである. 本実験で用いたブロック崩しの環境で可視化した戦略も, 同様に有効なものであったと考えられる. ブロック崩しの環境ではエージェントの行動はボールを跳ね返せるかどうかに直接関係しているが, ブロックの破壊を報酬とした場合, 報酬を得られる機会が少なかったり, エージェントの行動ともたらされる報酬の関係が間接的になることで, Montezuma's Revenge などの環境と同様に DQN のような手法では学習が進みにくいのに対し, 階層強化学習を用いることで効率的に学習することができたと考えられる. 今回用いた手法の課題としては, 階層強化学習のサブゴールがすでに決められたものの中から選ばれることがあげられる. このため, サブゴールを設定する際はエージェントが様々な戦略をとれるように十分な種類のサブゴールを選定する必要があると考えられる.

## 5. おわりに

実験により, 上位方策の決定を可視化することで AI のとっている戦略を説明できることが検証できた. 今後の課題として, 下位方策にルールベースのアルゴリズムを用いなかった場合や, より複雑な環境で実験した場合の有効性も検証したい.

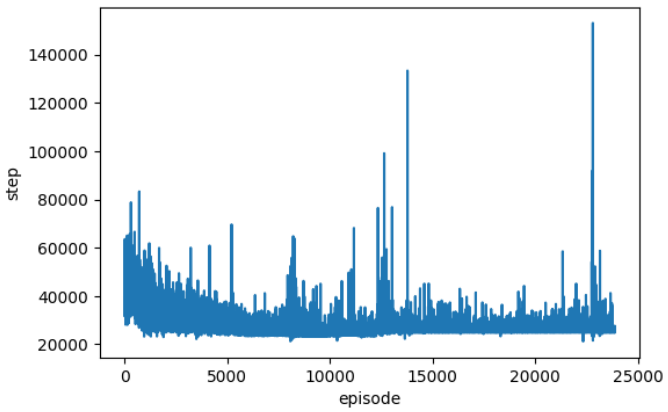


**図 9** 学習中のエピソードの長さ

### 参考文献

[1] Silver, D., et al.: Mastering the game of Go with deep neural networks and tree search, Nature, Vol. 529, No. 7587, pp. 484-489, (2016).

[2] 箭本進一: "[CEDEC 2019] ゲームの敵 AI は AI に作らせる。「『強い』を作るだけが能じゃない！ディープラーニングで 3D アクションゲームの敵 AI を作ってみた」聴講レポート", 4Gamer.net, 2019/9/6 更新, 最終閲覧日:2022/6/8, https://www.4gamer.net/games/999/G999905/20190906094

[3] 池田心:"楽しませる囲碁・将棋プログラミング", 2013/3/1,

©2022 Information Processing Society of Japan

「オペレーションズ・リサーチ: 経営の科学, 58 巻, 3 号, pp.167-173, http://hdl.handle.net/10119/12060
[4] 金子知適:“コンピュータ将棋を用いた棋譜の自動解説と評価”, 情報処理学会論文誌, Vol. 53, No. 11, pp. 2525–2532, (2012).
[5] 小田直輝, 永井秀利, 中村貞吾:“解説文生成における石の勢力を考慮した適切な囲碁用語の選択”, 2020 年度電気・情報関係学会九州支部連合大会（第 73 回連合大会）講演論文集, pp. 230-231, (2020).
[6] 松原仁:“速報　AlphaGo の勝利”, 情報処理, Vol. 57, No. 6, pp. 502–503, (2016).
[7] 谷岡広樹:”スポーツアナリティクスにおけるデータと AI 活用”, 教育システム情報学会誌, Vol. 37, No. 3, pp. 192-197, (2020)
[8] Breiman, L., Shang, N.: Born again trees. University of California, Berkeley, Berkeley, CA, Technical Report (1996)
[9] M.T. Ribeiro, S. Singh, C. Guestrin: “why should i trust you?”:Explaining the predictions of any classifier, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135-1144, (2016)
[10] S.M. Lundberg, S.-I. Lee: A unified approach to interpreting model predictions, Advances in Neural Information Processing Systems, pp. 4765-4774, (2017)
[11] P.W. Koh, P. Liang: Understanding black-box predictions via influence functions, Proceedings of the 34th International Conference on Machine Learning, PMLR, pp. 1885-1894, (2017)
[12] E. Angelino, N. Larus-Stone, D. Alabi, M. Seltzer, C. Rudin: Learning Certifiably Optimal Rule Lists for Categorical Data, Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 35–44, (2017)
[13] J. Bien, R. Tibshirani: Prototype selection for interpretable classification, The Annals of Applied Statistics 5, 4, pp. 2403-2424, (2011)
[14] Springenberg, J. T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806, (2014)
[15] M. Sundararajan, A. Taly, Q. Yan: Axiomatic attribution for deep networks, International Conference on Machine Learning, 70, JMLR. org, pp. 3319-3328, (2017)
[16] Greydanus S., Koul A., Dodge J., Fern A.: Visualizing and Understanding Atari Agents, In International Conference on Machine Learning, pp.1792–1801, (2018)
[17] Sutton, R. S., Precup, D. and Singh, S.: Between MDPs and semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning, Artificial Intelligence, Vol. 112, No. 1-2, pp. 181–211, (1999).
[18] Watkins, C., Dayan, P.: Q-Learning, Machine Learning, 8, pp.279-292, (1992)
[19] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D.: Human-level control through deep reinforcement learning, Nature, Vol. 518, No. 7540, pp. 529-533, (2015)
[20] Le, H., Jiang, N., Agarwal, A., Dudik, M., Yue, Y., Daumé III, H.,: Hierarchical imitation and reinforcement learning, In International conference on machine learning, pp.2917-2926, (2018)
[21] 高田司郎, 新出尚之: 意図に基づくエージェントアーキテクチャ (<特集>意図研究のスペクトル), 人工知能学会誌 20 (4), pp.433-440, (2005) ,
[22] 一杉裕志, 高橋直人, 中田秀基, 佐野崇: RGoal Architecture: 再帰的にサブゴールを設定できる H 強化学習アーキテクチャ, 人工知能学会第二種研究会資料, AGI-009, pp.05-, (2018)
[23] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347, (2017)
[24] Zahavy, T., Ben-Zrihem, N., Mannor, S.: Graying the black box:Understanding dqns, In International conference on machine learning, pp.1899-1908, (2016)

©2022 Information Processing Society of Japan 