情報処理学会論文誌 Vol.60 No.3 956–966 (Mar. 2019)

報酬が疎な環境に適した深層強化学習法

水上 直紀[1,a)] 鈴木 潤[2,b)] 亀甲 博貴[3,c)] 鶴岡 慶雅[4,d)]

受付日 2018年5月7日, 採録日 2018年12月4日

概要：深層強化学習の発展により，ゲーム画面とスコアの情報のみから，熟練のプレイヤを超えるスコアに到達可能なビデオゲームの自動操作エージェントを学習可能であることが示された．しかし，これまであまり言及されていないが，報酬が疎な環境のゲームに関しては，いまだ人間のスコアにはるかに及ばない．これは，一般的な強化学習の枠組みそのものが，報酬が疎な環境からの学習に適していない点に起因すると考えられる．そこで本論文では，ビデオゲームの自動操作エージェントを学習するタスクを題材に，報酬が疎な環境でも効果的にエージェントの学習が可能となる枠組みについて議論を行い，報酬が疎な環境に適した深層強化学習の枠組みを提案する．ビデオゲームの自動操作エージェントの性能を測るベンチマークとして用いられる Atari 2600 において報酬が疎な環境となるゲームを用いて提案手法の評価実験を行い，提案手法が，従来の一般的な深層強化学習法を大きく上回るスコアを達成できることを示す．

キーワード：深層強化学習，探索，自動操作エージェント

Deep Reinfocement Learning for Sparse Reward Environments

Naoki Mizukami[1,a)] Jun Suzuki[2,b)] Hirotaka Kameko[3,c)] Yoshimasa Tsuruoka[4,d)]

Received: May 7, 2018, Accepted: December 4, 2018

Abstract: Recent deep reinforcement learning (RL) algorithms demonstrate that they can successfully learn super-human-level video game agents only from the information on game screens and scores. However, a closer look at their performance reveals that the algorithms fall short of humans in games where rewards are only obtained occasionally. This is mainly because the RL framework does not fit such sparse reward environments. From this perspective, we discuss how we can build agents that specifically fit the sparse reward environments, and propose an effective method. We evaluate our method on Atari 2600 games with sparse rewards, and report that our method can provide significant improvements over conventional RL methods.

Keywords: deep reinforcement learning, exploration, video game agent

1. はじめに

ロボット，自動車，ドローンといった実世界の機械を制御する方法を自動的に獲得する技術として，深層強化学習技術が広く注目されるようになった．しかし，実環境での実証実験には費用や時間のコストが非常にかかるという大きな課題があるため，基盤となる深層強化学習技術の開発は，将棋/囲碁/ビデオゲームといった明確なルールと仮想的な環境が存在するゲームを用いて行われることが多い．たとえば，Arcade Learning Environment（ALE）[1] に実装されている Atari 2600 のゲームは，深層強化学習法のベンチマークとしてよく用いられている．こういった仮想環境において，事前にそのゲームのルールを与えられていない状態から[*1]，ゲームの画面（まはた局面）とスコア（または勝敗）のみの情報を用い，実際にプレーしながらその環境（ゲーム）に適したエージェントの操作戦略を自動で構

1 HEROZ 株式会社
HEROZ, Incorporated, Minato, Tokyo 108–0014, Japan
2 東北大学大学院情報科学研究科
Graduate School of Information Science, Tohoku University, Sendai, Miyagi 980–8579, Japan
3 京都大学学術情報メディアセンター
Academic Center for Computing and Media Studies, Kyoto University, Kyoto 606–8315, Japan
4 東京大学工学系研究科
Graduate School of Engineering, The University of Tokyo, Bunkyo, Tokyo 113–8654, Japan
a) mizukami@heroz.ac.jp
b) jun.suzuki@ecei.tohoku.ac.jp
c) kameko.hirotaka.6n@kyoto-u.ac.jp
d) tsuruoka@logos.t.u-tokyo.ac.jp

[*1] 合法手/非合法手程度の最低限の情報は与えられると仮定．

築する，という最終的には実環境への適用へつながるタスク設定がしばしば深層強化学習の研究に用いられている．

一般論として，Deep Q-Network（DQN）[2] といった近年開発された深層強化学習法を用いて構築された Atari 2600 中の各ゲームをプレーするエージェントは，ゲーム画面とスコアの情報のみから学習を行い，人間の平均的なプレーヤーのレベルを超えているといわれる．しかし実際は，強化学習によって構築されたエージェントがうまく機能するかどうかは，ゲームの性質によって大きく異なる．Bellemare らは，文献 [3] の中で，強化学習のアルゴリズムにとって，「良い報酬が得られる局面を探索するのが容易か」という観点で各ゲームの難易度を分類している．具体的には，探索が容易と考えられるゲームを**探索容易（easy exploration）ゲーム**，困難と考えられるゲームを**探索困難（hard exploration）ゲーム**と呼ぶ．また，「報酬が得られる局面が多いか」という観点で探索困難ゲームをさらに 2 種類に分類している．探索困難ゲームの中で，報酬が得られる局面が多いゲームを**密な報酬環境（dense reward）ゲーム**，報酬が得られる局面が少ないゲームを**疎な報酬環境（sparse reward）ゲーム**と呼んでいる．

探索容易ゲームでは，エージェントに ϵ-greedy といった簡単な探索戦略を適用するだけで，高いスコアを比較的容易に獲得できる．一方，探索困難ゲームでは，ϵ-greedy といったランダム性に基づく方法では不十分であり，何かしらの高度な探索戦略が必要と考えられる．ただし，探索困難ゲームでも密な報酬環境ゲームに関しては，最近の深層強化学習技術の発展にともない，平均的な人間レベルに到達している [4], [5], [6], [7]．しかしながら，疎な報酬環境ゲームにおいては，最新の深層強化学習の成果を適用しても平均的な人間レベルには至っていない．その理由として考えられることとして，たとえば強化学習を行う際に，報酬が得られる局面を見つけられなかった場合，強化学習の方法がどんなに優秀だとしても学習を行うことができないのは明らかである．つまり，疎な報酬環境ゲームでは，何かしらの方法を用いて報酬が得られる局面を見つけ出さなくてはいけない．しかし，ゲーム画面とスコアのみからエージェントを構築する設定では，Atari 2600 のゲームは列挙困難なほど大きい局面空間を持つため，その探索は容易ではない．

そこで本論文では，巨大な局面空間を持つ疎な報酬環境ゲームでも有効に機能する方法論について議論し，こういった状況に適したを考案する．まず提案手法は，ベースの深層強化学習法として，汎用的かつ学習性能の高い Asynchronous Advantage Actor-critic（A3C）[8] アルゴリズムを採用する．そして，ベースとなる A3C に対して疎な報酬環境ゲームに適した改良を行うことで提案手法を構築する．また，提案手法の効果を検証するため，A3C が人間と比較し苦戦している疎な報酬環境ゲームに着目し，それらを用いて実証実験を行い，提案手法が A3C と比較して疎な報酬環境ゲームのスコアは大きく上昇可能であること，さらに一部のゲームにおいて最高スコアを達成可能であることを示す．

2. 問題設定

強化学習の枠組みでは，エージェントが行動し，その行動に対する報酬に基づいてエージェントの学習が行われる．報酬が疎とは，ある局面（状態）に対する行動に与えられる報酬が大半の場合でゼロ[*2]であることをさす．$\mathcal{S}$ を，すべての可能な局面の集合を表す記号とする[*3]．また，$\mathcal{A}$ をすべての可能な行動の集合とする．次に，t を時刻を表す変数とし，$t \in \{0, 1, 2, \dots\}$ のように 0 および正の整数を取ることとする．このとき，$s_t \in \mathcal{S}$ を時刻 t の局面，$r_t \in \mathbb{R}$ を時刻 t の報酬，$a_t \in \mathcal{A}$ を時刻 t の行動を表す記号とする．

対象とする環境がマルコフ決定過程に従うと仮定した場合，報酬 r_{t+1} は報酬確率 $p(r_{t+1} \mid s_t, a_t, s_{t+1})$ に基づいて与えられる．ここで，すべての可能な (s_t, a_t, s_{t+1}) の組合せの集合を $\mathcal{T}$ とおく．このとき，本論文では，「報酬が疎な環境」という用語を以下のように定義する．

定義 1（報酬が疎な環境）．ある特定の (s_t, a_t, s_{t+1}) の組合せの集合を $\mathcal{T}' \subset \mathcal{T}$ とする．このとき $|\mathcal{T}'| \ll |\mathcal{T}|$ と仮定する．すなわち，特定の組合せ $\mathcal{T}'$ は，全体に対して要素数が非常に少ない集合である．このとき，すべての $(s_t, a_t, s_{t+1}) \in \mathcal{T} \backslash \mathcal{T}'$ に対して，$r_{t+1} = 0$ となる確率が 1 となる環境，すなわち

$$p(r_{t+1} = 0 \mid s_t, a_t, s_{t+1}) = 1 \quad \forall (s_t, a_t, s_{t+1}) \in \mathcal{T} \backslash \mathcal{T}' \tag{1}$$

を，報酬が疎な環境と呼ぶ．

この環境の意味するところは，どのような局面から次の局面へ遷移する行動をとっても，報酬を得られる確率が限りなく低いことを表している．このような環境の場合，報酬が得られる行動の系列が得られることが少ないため，エージェントの学習を進めることが困難であることを意味している．

表 1 に本論文で用いる記号の一覧を示す．

3. 関連研究

3.1 深層強化学習

Mnih ら [2] はディープニューラルネットワークを利用して，Q 関数を学習する DQN と呼ばれる手法を提案した．この手法により人間が観測するゲームの画面情報と現在のスコアのみから Q 関数を効果的に直接学習することに成功

[*2] 報酬は一般的に正の報酬のみを用いることが多いが，負の報酬がある場合も含むこととする．

[*3] ここでの「局面」とは，「状態」とも呼ばれるが，一般名詞としての「状態」との混同を避けるために本論文では局面という用語を用いる．

表 1 本論文で用いられる記号
Table 1 A list of notations used in this paper.

t	時刻．$t \in \{1, 2, \dots\}$
$\mathcal{S}$	すべての可能な局面の集合
$\mathcal{A}$	すべての可能な行動の集合
a または a_t	行動，または，時刻 t での行動．$a_t \in \mathcal{A}$
s または s_t	局面，または，時刻 t の局面．$s_t \in \mathcal{S}$
r または r_t	報酬，または，時刻 t での報酬，探索報酬との区別のために「真の報酬」という記載も用いる
e または e_t	探索報酬，または，時刻 t での探索報酬
$p(r_{t+1} \mid s_t, a_t, s_{t+1})$	局面 s_t から s_{t+1} へ行動 a_t により遷移した際の $t+1$ での報酬 r_{t+1} の確率
$\mathcal{T}$	(s_t, a_t, s_{t+1}) の組合せの集合
π または $\pi(a_t \mid s_t; \theta'_\pi)$	戦略，または，パラメタ θ'_π のときに局面 s_t で行動 a_t を選択する確率
v または $v(s_t; \theta'_v)$	価値関数，または，パラメタ θ'_v のときに局面 s_t の価値を表す値
θ，または，θ_π，θ_v	パラメタ，または，戦略 π に関連するパラメタ，および，価値関数 v に関連するパラメタ
θ'，または，θ'_π，θ'_v	パラメタ，または，戦略 π に関連するパラメタ，および，価値関数 v に関連するパラメタ（θ と比較した場合は，各スレッドごとのローカルなパラメタを表す）．
$H(\theta'_\pi)$	エントロピー項
$\mathcal{Z}$	任意の有限の整数空間．$\mathcal{Z} = \{1, \dots, N\}$
N	局面のハッシュ値の上限
$\phi(s_t)$	ハッシュ関数 $\phi : \mathcal{S} \to \mathcal{Z}$ 例：$\phi(s_t) = z$，ただし $s_t \in \mathcal{S}$ および $z \in \mathcal{Z}$．
z_t	時刻 t の局面 s_t からハッシュ関数 $\phi : \mathcal{S} \to \mathcal{Z}$ により変換したハッシュ値
$Z_{1:t}$	時刻 1 から時刻 t の局面 s_t からハッシュ関数 $\phi : \mathcal{S} \to \mathcal{Z}$ により変換したハッシュ値
$\psi(Z_{1:t}, z_t)$	整数のリスト $Z_{1:t}$ に対して整数 z_t の出現回数を返す関数
$\tilde{\psi}(s_t)$	現在の局面 s_t の擬似出現回数を計算する関数
β	項の重み係数，β^{a3c}，β^{hash}，β^{psc}，β^{ucb} など
γ	時刻ごとの報酬の割引率

した．

Mnih ら [8] は，さらに Actor-Critic 法に基づく A3C アルゴリズムを提案した．A3C は，Atari 2600 を用いた実験において高い性能を示した代表的な深層強化学習法であり，DQN よりも探索困難ゲームを除けば優れている．また，A3C は，DQN のように Q 関数を学習するのではなく，行動戦略と価値関数を分離して同時に学習を行う方式を採用している．このような学習方式は行動空間の高度な表現力を持つ一方，表現力の向上による過学習になりやすい傾向があり，学習時の探索が局所的になる可能性が高い．この問題を解決するため，A3C では目的関数にエントロピー正則化項を加え過学習を軽減する仕組みを導入している．より具体的には，エントロピー正則化項は，すべての行動を同じ確率で行うとき，すなわち，エージェントがランダムに行動する際に最も値が高くなる．よって，エントロピー正則化項を追加することで，エージェントは報酬が得られた行動のみをとるように学習が進むといった過学習を軽減することができる．

Algorithm 1 各 actor-learner スレッドごとの A3C の学習アルゴリズム

```
// T：共有するグローバル変数（全体で T = 0 で初期化）
// θ_π，θ_v：共有するグローバル変数，θ'_π，θ'_v：スレッド固有の変数
// t_interval：更新間隔パラメータ，T_max：最大繰返し回数
初期化 t ← 1
repeat
    dθ_π ← 0, dθ_v ← 0, θ'_v ← θ_v, θ'_π ← θ_π        //毎回初期化
    t_s ← t
    repeat
        戦略 π(a_t|s_t; θ'_π) から選択された行動 a_t
        得られる報酬 r_t と次局面 s_{t+1}
        t ← t + 1
        T ← T + 1        //すべてのスレッドについて非同期に更新
    until s_t = 終端局面 Or t − t_s = t_interval
    if s_t = 終端局面 then
        R ← 0
    else
        R ← v(s_t; θ'_v)
    end if
    foreach i ∈ {t − 1, . . . , t_s} do
        R ← γR + r_i
        dθ_π ← dθ_π + dθ'_π 式 (2) より θ'_π の累積勾配 dθ_π
        dθ_v ← dθ_v + dθ'_v 式 (3) より θ'_v の累積勾配 dθ_v
    end for
    dθ_π と dθ_v を用いた θ_π と θ_v の非同期更新
until T ≥ T_max
```

Algorithm 1 に A3C の学習手順の概要を示す．R を特定の周期で観測された報酬の加重和とする．π と v はそれぞれ，戦略と価値関数の出力と表す．θ_π と θ_v はそれぞれ π と v のパラメタである．同様に θ'_π と θ'_v はそれぞれ π と v のパラメタであり，各スレッドごとに管理されるスレッドごとに独立なパラメタとする．さらに $H(\cdot)$ はエントロピー項，β^{a3c} はエントロピー項の強さを決定する係数である．このとき，A3C は各スレッドの独立なパラメタに対して勾配 $d\theta'_\pi$ と $d\theta'_v$ を次の計算によって求める．

$$d\theta'_\pi = \nabla_{\theta'_\pi} \log(\pi(a_i|s_i;\theta'_\pi))(R - v(s_i;\theta'_v)) + \beta^{\mathrm{a3c}} \nabla_{\theta'_\pi} H(\theta'_\pi), \tag{2}$$

$$d\theta'_v = \frac{\partial (R - v(s_i;\theta'_v))^2}{\partial \theta'_v} \tag{3}$$

これらの値を用いて最終的にはグローバルなパラメタ θ_π と θ_v を更新する．

3.2 強化学習における探索と探索報酬

報酬が疎な環境に対応する方法の 1 つとして MBIE-EB アルゴリズム [9] が提案されている．MBIE-EB アルゴリズムは，局面の訪問回数を記録し，その局面の訪問回数に基づく探索報酬を報酬の一部として利用することで，未知の局面へより遷移しやすくなるように学習を進めて報酬が得られる局面を探索する方法となっている．この方法は，局面空間が列挙可能な程度に小さいマルコフ決定過程の設定では有効であると実証されている．しかし，近年さかん

に取り組まれている深層強化学習の枠組みでは，ゲーム画面のピクセル情報をそのまま入力情報として用いている．これは，すべての可能なゲーム画面の集合により局面空間 $\mathcal{S}$ が定義されることを意味する．たとえば，文献 [2] では，各ゲーム画面は 84×84 ピクセルの 256 階調で構成され，さらに 4 つの連続する時刻の画面を 1 つの入力として利用している．つまり，単純計算で最大 $256^{84\times84\times4}$ 種類の局面が存在する環境と解釈できる．このように膨大な局面数となる環境下では，同一の局面に到達する確率は非常に小さい．このため，単純に各局面の出現回数を手がかりとして探索する MBIE-EB アルゴリズムのような方法は機能しないことが容易に推測できる．

この状況に対応するため，複数の局面の抽象化手法が深層強化学習の分野において提案されている．たとえば，Tang ら [10] は，ハッシュ関数による局面の抽象化手法を用いて，以下のような探索報酬 e^{hash} を提案している．

$$e_t^{\text{hash}} = \frac{\beta^{\text{hash}}}{\sqrt{\psi(Z_{1:t}, z_t)}}, \tag{4}$$

ここで $\beta^{\text{hash}} \in \mathbb{R}_{\geq 0}$ は探索報酬の係数である．局面の集合 $\mathcal{S}$ から有限の整数空間 $\mathcal{Z}$，つまり $\mathcal{Z} = \{1, \ldots, N\}$ に写像するハッシュ関数を $\phi : \mathcal{S} \to \mathcal{Z}$ で定義する．たとえば $z_t \in \mathcal{Z}$ を，ハッシュ関数 ϕ によって時刻 t の局面 $s_t \in \mathcal{S}$ から ϕ によって変換された整数とする．次に，$Z_{1:t}$ を，z_1 から時刻 z_t までを並べた整数のリストとする．つまり，$Z_{1:t} = (z_1, \ldots, z_t)$ である．このとき，$\psi(Z_{1:t}, z)$ を整数のリスト $Z_{1:t}$ に対して整数 z の出現回数を返す関数とする．たとえば，$Z_{1:6} = (1, 2, 1, 3, 3, 1)$ および $z_t = 1$ のとき，$Z_{1:t}$ 中に 1 が 3 回出現するので $\psi(Z_{1:t}, z_t) = 3$ となる．$\psi(Z_{1:t}, z_t)$ は Z_t の長さに対して，単調非減少関数なので，時刻 t が大きくなれば e^{hash} の値は徐々に小さくなっていく．また，z_t と同じ整数が出現すればするほど値が小さくなる関数なので，式 (4) は，同じ局面を経験するたびに探索報酬が減少する関数と解釈することができる．

別の局面の抽象化手法として，Bellemare ら [3] は，局面の疑似出現回数を用いた探索手法を提案した．探索報酬 e^{psc} は次の式で定義される．

$$e_t^{\text{psc}} = \frac{\beta^{\text{psc}}}{\sqrt{\tilde{\psi}(s_t) + 0.01}}, \tag{5}$$

このとき，β^{psc} は探索報酬の係数である．局面 s_t の疑似出現回数 $\tilde{\psi}(s_t)$ は，現在の局面 s_t に対して，局面の疑似回数の出現確率 $\rho(s_t)$ と，次に同じ局面に遭遇したと仮定した際の出現確率 $\rho'(s_t)$ を用いて以下の式で計算する．

$$\tilde{\psi}(s_t) = \frac{\rho(s_t)(1 - \rho'(s_t))}{\rho'(s_t) - \rho(s_t)}, \tag{6}$$

この手法は特に “Montezuma’s revenge” という Atari 2600 の中でも深層強化学習アルゴリズムにとって最も難しいゲームの 1 つの成績を大きく向上した．

Tang らや Bellemare らの手法は局面の未知さを報酬とするものであるが，Intrinsic Curiosity Module（ICM）[11] では，局面遷移の予測の難しさを「好奇心」とすることで報酬の自動生成を行う．この報酬の生成には逆モデルと順モデルの 2 つのモデルが関係している．逆モデルでは観測された局面 s_t，s_{t+1} からエージェントの行動 a_t の予測を行う．順モデルでは局面 s_t と行動 a_t から次の局面 s_{t+1} の予測を行う．次にそれぞれのモデルの予測と実際の行動や局面から誤差を計算する．その誤差が大きいということはエージェントが理解していない行動や局面であり，好奇心の大きさを表現している．すなわちその誤差を報酬とすることでエージェントは好奇心に基づいた未知の局面を探索する行動をとることができる．この手法により VizDoom と Super Mario Bros. において外部の報酬を用いることなく，効率的な探索を行うことに成功した．

3.3 探索における選択基準

Lai と Robbins [14] は，Multi Armed Bandit（MAB）問題に対して Upper Confidence Bounds（UCB）スコアが最大となるよう行動を選択するというアルゴリズムを提案した．このアルゴリズムは異なる行動どうしの関係を考慮するために全体の試行回数を用いて行動を決定する．UCB スコア r^{ucb} は以下の式で定義される．

$$r_t^{\text{ucb}} = r(a_t) + \sqrt{\frac{2\log(t)}{\psi(A_{1:t}, a_t)}}, \tag{7}$$

a_t は時間 t で選択された行動，$A_{1:t}$ は時刻 1 から t までの行動の履歴，$r(a_t)$ は行動 a_t をとった際の平均報酬，$\psi(A_{1:t}, a_t)$ は行動 a_t がこれまでに選択された回数を表すとする．UCB スコアの第二項は各行動を選択する情報の価値を表している．UCB アルゴリズムは無限回の試行で最良の行動を選択できると保証されている．

3.4 報酬が疎な環境における学習

本論文が対象としているビデオゲームの自動操作エージェントを学習するタスク以外でも，たとえばロボットアームの研究分野においても同様に報酬が疎な環境を対象にした研究が行われている．ロボットアームの課題として箱を開け，中にブロックを配置する課題などは，報酬を得るための適切な行動は複雑であり，ランダムな行動で報酬を得ることは非常に困難である．そのためロボットアームの分野においても報酬が疎な環境において強化学習を行うことは挑戦的な課題である．

Andrychowics ら [12] は Hindsight Experience Replay と呼ばれる疑似ゴールを設定することで効率的な学習に成功した．この疑似ゴールは報酬の得られなかったエピソードから生成される．そしてこの疑似ゴールに対する行動と局

面をセットにして価値関数を学習する．

Riedmiller ら [13] は意図戦略を定義することで本来のゴールを学習する Scheduled Auxiliary Control を提案した．意図戦略はゴールからの距離に応じて与えられる報酬をもとに学習が行われる．またスケジューラが個別の意図戦略を選択し実行することでより効率的な学習が行われる．

これらの論文と本論文では問題設定が異なる．ロボットアームの分野ではたとえばゴールからの距離といった単純で直感的な疑似報酬を設計することができる．一方，Atari2600 のゲームでは，どのような状態のときに報酬が与えられると考えていいのかは自明ではない．そのため，提案手法では「未知の局面」に対して報酬を設計しているという点が大きく異なる．

4. 報酬が疎な環境に適した深層強化学習法

ここでは提案手法について説明する．**図 1** に，提案手法における各モジュールとモジュール間の関係を示す．

提案手法は，前章で説明した A3C をベースの深層強化学習法として採用する．その上で，ベースの A3C から報酬が疎な環境でも学習を効率的に行うために以下の 3 点に関して改良を行う．

- UCB に基づく探索報酬を計算するモジュールの追加（4.1 節）
- 探索用の戦略（探索戦略）と実行用の戦略（活用戦略）という 2 つの異なる目的で使用される戦略の組み込み（4.2 節）
- 報酬が得られた行動の再利用（4.3 節）

4.1 UCB に基づく探索報酬

式 (4) や式 (5) に示した従来法の探索報酬は，同一の局面に遭遇する度に値が単調に減少する関数である．つまり，未知の局面をより優先的に選択する戦略といえる．しかし，報酬が疎な環境では，未知の局面を優先する戦略，逆にいえば，既知の局面をなるべく選択しないようにする戦略が良いとは必ずしもいえない．これは，報酬が疎な環境では，エージェントが報酬を得られる局面に仮に近づいていたとしても，報酬が得られる局面まで必ず到達できる保証がないため，同じ局面をある程度の試行を繰り返す必要があるからである．この仮説に基づき，我々は式 (7) で紹介した UCB に基づく探索報酬を提案する．

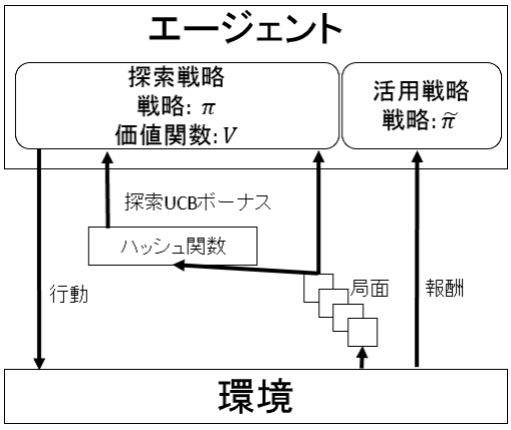


図 1 提案手法における各モジュールの関係
Fig. 1 Relation among modules in the proposed method.

時刻 t における探索報酬を e_t とする．このとき，e_t を以下のように定義する．

$$e_t = \beta^{\mathrm{ucb}} \sqrt{\frac{\log(t)}{\psi(Z_{1:t}, z_t)}}, \tag{8}$$

ただし，β^{ucb} は e_t の重みを調整する係数とする．

e_t は，式 (7) に示した UCB の計算式の右辺第二項を元に定義されている．しかし，提案手法の状況に合うように修正を行っている．具体的には，本来の UCB は行動回数 $\psi(A_{1:t}, a_t)$ に基づいて値を計算するが，e_t は，その代わりに局面の訪問回数 $\psi(Z_{1:t}, z_t)$ に基づいて値を計算する点が異なる．

e_t の直観的な解釈としては以下のようになる．まず基本的に，ある時刻 t を固定した場合，ある局面 s_t に対するハッシュ関数 $\phi(s_t)$ の値 z_t の出現回数が小さければ e_t は大きい値になり，反対に，z_t の出現回数が大きければ e_t の値は小さくなる．つまり，頻度に基づく方法と同様に，基本的には未知の局面を優先的に探索する方法となっている．一方で，z を固定して考えた場合，時刻 t が進むに連れ e_t の値は対数スケールで徐々に大きくなる．つまり，長期間 z の局面を訪問しなかった場合，改めて z の局面を選択するようになる可能性が高まることを意味する．このように，式 (8) に基づく探索報酬は，訪問回数に基づいて未知局面を優先して選択するだけでなく，時間変化による探索の進み具合も考慮した計算式になっているといえる．これ以降，本論文で探索報酬と記述する際には式 (8) で定義された UCB に基づく探索報酬のことを指すこととする．

4.2 2 種類の戦略の学習

この章では，提案手法の全体的な学習手順について説明する．Algorithm 2 は提案手法によるエージェントの学習方法を示しており，基本的にはベースラインの A3C と同じ手法である．最初にエージェントは，現在の戦略に基づいて行動を選択し，次の局面，報酬，および，探索報酬を受け取る．提案手法が，A3C や他の同様の探索方法を用いる手法と大きく異なる点は，真の報酬と探索報酬を区別して 2 つの戦略を独立に学習することである．

以下，2 つの戦略を用いる理由を述べる．一般的に，これまでの関連研究では探索報酬 e は真の報酬 r に追加して使用される．すなわち，$R = r + e$ とおき，R はエージェントのネットワークを更新するために r の代わりに用いられる．これらの方法では，戦略と価値関数の適切な訓練とい

Algorithm 2 各 actor-learner スレッドごとの提案手法 (A3C+UCB) の学習アルゴリズム

```
// T：共有するグローバル変数（全体で T = 0 で初期化）
// θ_π̃, θ_π, θ_v：共有するグローバル変数
// θ'_π̃, θ'_π, θ'_v：スレッド固有の変数
// t_interval：更新間隔パラメータ，T_max：最大繰返し回数
初期化：t ← 1
repeat
    dθ_π̃ ← 0, dθ_π ← 0, dθ_v ← 0, θ'_π̃ ← θ_π̃, θ'_π ← θ_π, θ'_v ← θ_v
    t_s ← t
    repeat
        π̃(a_t|s_t; θ'_π̃) に基づく行動 a_t.
        観測された r_t と次局面 s_{t+1}
        t ← t + 1
        T ← T + 1        //すべてのスレッドについて非同期に更新
    until s_t 終端局面 Or t − t_s = t_interval
    if s_t = 終端局面 then
        R ← 0
    else
        R ← v(s_t; θ'_v)
    end if
    foreach i ∈ {t − 1, . . . , t_s} do
        R ← γR + e_i
        R' ← γR' + r_i
        R̃ ← clip(R', −1, 1) //R' を [−1, 1] の範囲でクリップ
        dθ_π̃ ← dθ_π̃ + dθ'_π̃ 式 (9) より θ'_π̃ の累積勾配 dθ_π̃
        dθ_π ← dθ_π + dθ'_π 式 (2) より θ'_π の累積勾配 dθ_π
        dθ_v ← dθ_v + dθ'_v 式 (3) より θ'_v の累積勾配 dθ_v
    end for
    dθ_π̃ と dθ_π と dθ_v を用いた θ_π̃ と θ_π と θ_v の非同期更新
until T ≥ T_max
```

う観点で問題を引き起こす可能性がある．これは，R は r と e の和で表されるため，エージェントはどちらの報酬に基づいて現在の学習が行われているか明確に分かっていない状態で学習が行われることを意味する．つまり，報酬が疎な環境では，エージェントは探索報酬に従って探索を行うのみの学習となっている可能性が高く，実際に報酬が得られた際の価値を適切に判断していない可能性がある．また，学習時の探索報酬 e は，時刻 t が無限になるまで学習を行った場合，理論的には 0 に近似するので，最終的に探索報酬は真の報酬の学習には悪影響を与えない．しかし，学習時の更新回数は有限であるため，現実的な状況では探索報酬 e_t は 0 にはならない．さらに，真の報酬 r はほぼ 0 であるため，学習される戦略は，結局探索報酬の影響を強く反映した戦略となる可能性が高い．

上記の推測に基づいて，報酬と探索報酬用の 2 つの異なる戦略を導入する．1 つ目の戦略は**探索戦略** π と呼び，トレーニング時に探索報酬を用いて学習する．2 つ目の戦略は**活用戦略** $\tilde{\pi}$ と呼び，環境からの真の報酬のみで学習し，主にテスト時に使用する．これらの 2 つの戦略は別々の報酬で学習を行うが，報酬や探索報酬に関する情報を共有させるため単一の畳み込みニューラルネットワークを使用する．

$\theta'_{\tilde{\pi}}$ を $\tilde{\pi}$ を計算する際に利用されるパラメタの集合とする．式 (2) および式 (3) と同様に，$d\theta'_{\tilde{\pi}}$ は以下の式を用いて計算する．

$$d\theta'_{\tilde{\pi}} = \nabla_{\theta'_{\tilde{\pi}}} \log(\tilde{\pi}(a_i|s_i;\theta'_{\tilde{\pi}}))\tilde{R} \tag{9}$$

4.3 経験再生

これまで述べてきたように，報酬が疎な環境では，真の報酬が得られる局面に到達する可能性が非常に小さいという問題がある．この問題から，エージェントの学習途中で運良く報酬が得られる局面に到達したとしても，報酬が得られた局面を再現することが難しいということが容易に想像できる．これは，エージェントは報酬を一度受け取っても，パラメータの変動幅は小さいため，同じ行動を再現できるようになるまでには繰返しの学習が必要となる．この課題に対応するため以前の研究では経験再生と呼ばれる手法が用いられている [2]．本研究ではエージェントが真の報酬を得た際の局面と行動を履歴として記録し，それを再利用する枠組みを導入する．

この履歴をトレーニング時に ϵ の確率で用いる．これによりエージェントに何度も同じ行動を強制的にとらせることで学習を効果的に行う．本研究では ϵ の値を 0.1 とした．

5. 実験

本論文では，ALE [1] に実装されている Atari 2600 のゲームを用いて実験を行った．

5.1 疎な報酬環境ゲーム

文献 [3] で疎な報酬環境ゲームに分類されているゲームを対象に実証実験を行う．具体的には，“Freeway”，“Gravitar”，“Montezuma revenge（Montezuma）”，“Private Eye（Private）”，“Solaris”，“Venture” の 6 種類のゲームである．また，文献 [3] では報酬が疎なゲームに分類されていないが，本研究には 1 章で述べたように A3C の改良という観点もあるため，ベースラインの A3C のスコアが 0 であった “Enduro” も実験に追加し，報酬が疎なゲームと同様に，探索によって報酬が容易に得られるようになるか効果を検証する．[*4] よって，合計 7 種類のゲームを用いて実験を行った．

[*4] Enduro は，いわゆる「車のレースゲーム」であり，車を抜かすたびにスコアが加算されるので，その観点では疎な報酬環境ゲームではない．しかし，基本操作として，「車のアクセル」に相当する操作を選び続けるという行為が学習できるまでは，報酬を一度も得られないまま試行が終了することがほとんどとなる．よって，学習がうまくいっていない段階では，報酬が疎なゲームで解消したい状態と同じと見なせる．実際，ベースラインの A3C では，2 億ステップの学習が終わった状況でも「車のアクセル」に相当する操作を選択し続ける．ということを学習できていないため，一度も報酬が得られず，スコアが 0 点のままとなる．

表 2 トレーニングとテストに用いられる構成の概要
Table 2 Summary of configurations used for training and evaluation.

学習時	トレーニングステップ	2 億ステップ（8 億フレーム）
	スレッド数	56
	最適化手法	RMSProp，減衰率 0.99
	割引率 γ	0.99
	エントロピー項の係数 β^{a3c}	0.01
評価時	テスト間隔	100 万ステップごと
	テスト回数	それぞれ 30 回

Algorithm 3 局面 s からハッシュキー z を計算するアルゴリズム

Input: 局面 s
// k：ハッシュの粒度を表す変数
// $\boldsymbol{A} \in [-1,1]^{k \times D}$：ランダム写像行列
// $g(\cdot)$：局面 s を D 次元ベクトルに写像する関数
// p_r：インデックスの最大値を決定する素数
$z \leftarrow 0$ // z：ハッシュキーを計算するための変数
$\tilde{\boldsymbol{b}} \leftarrow g(s)$ // $\tilde{\boldsymbol{b}}$：D 次元ベクトル
$\boldsymbol{b} \leftarrow \boldsymbol{A}\tilde{\boldsymbol{b}}$ // $\boldsymbol{b}$：k 次元ベクトル
for $i = 1$ to k **do**
  $z \leftarrow z \times 2$
  if $b_i > 0$ **then** // $\boldsymbol{b} = (b_1, \ldots, b_k)$
    $z \leftarrow z + 1$
  end if
end for
$z \leftarrow z \% p_r$
Output: z

5.2 ベースライン手法の設定

本論文では，文献 [8] 中の A3C の実験で使用された実験設定に従って実験を行った[*5]．たとえば，ネットワークアーキテクチャは既存研究 [15] と同一である．具体的には，2 層の畳み込みニューラルネットワーク（スライド 4 で 8×8 の 16 枚のフィルタとスライド 2 で 4×4 の 32 枚のフィルタ）と 256 の隠れユニットを持つ全結合層の 3 層からなる構成である．隠れユニットの活性化関数として，ReLU [16] を使用した．また，最終出力層は，価値関数と各行動ごとの確率を出力するソフトマックス層の 2 種類で構成される．

表 2 に，学習および評価時の設定の概要を示す．まず，既存の研究と一致させるために，エージェントの学習ステップ数を 2 億ステップとした．次にエージェントの評価時には，“no-op performance measure” [4] という方法でさまざまな初期ランダム条件で 30 回評価を行った．最終的な評価は，30 回の平均スコアを最終スコアとして用いた．

5.3 ハッシュ値の計算方法

ここで，提案手法における局面 s からハッシュ値 z を計算する具体的な方法を述べる．実際に用いた処理手順を Algorithm 3 に示す．Algorithm 3 は，基本的に Locality-Sensitive Hashing（LSH）[17] と呼ばれる計算法であり，本質的には Tang ら [10] の研究で用いられている計算方法と同じである．

まず初期化処理として，各要素を $[-1,1]$ の乱数で初期化した $k \times D$ 行列 $\boldsymbol{A}$ を用意する．次に前処理として，局面 s を関数 $g(\cdot)$ を用いて D 次元ベクトル $\tilde{\boldsymbol{b}}$ に変換する．前述のとおり文献 [2] に従うと，各ゲーム画面は 84×84 ピクセルで構成され，さらに 4 つの連続する時刻の画面を 1 つの入力として利用する．提案手法では，関数 $g(\cdot)$ を，4 つの連続する画面の最後の画面と 1 番目と 4 番目の画面の差分という 2 つ分の画面を考え，それを一次元のベクトルに連結して返す関数と定義する．つまり，$D = 14{,}112$（$= 84 \times 84 \times 2$）である．次にベクトル $\tilde{\boldsymbol{b}}$ を，変換行列 A によって k 次元ベクトル $\boldsymbol{b}$ に写像する．最後に，$\boldsymbol{b}$ の各要素に対して，正の値の場合は 1，それ以外の場合は 0 として，$\boldsymbol{b}$ からハッシュキー（バイナリコード）z を得る．

LSH において k の値は粒度を決定する．値が大きいほど衝突が少なくなるため，局面を一意に区別することができる．理想的には現在とその後の局面のハッシュキーはつねに異なることが望ましい．そのためには k を大きな値にする必要がある．しかしながら，メモリ要求は k の値に比例して増加する．そこで必要なメモリ所要量を減らすため，インデックスの最大数を表す素数 p_r を使用する．ここでは Tang ら [10] の研究を参考にして $p_r = 999983$，$k = 128$ とした．

6. 結果および考察

6.1 結果

表 3 に実験結果を示す．議論の都合上，実験結果を 4 つのカテゴリ (a)，(b)，(c)，(d) に分類する．カテゴリ (a) は本論文で実際に実装して実験を行った結果である．このカテゴリの主な目的は，ベースライン手法（A3C），ベースラインに既存の探索法を加えたもの（A3C + psc），および提案手法（A3C + UCB）を公平な条件で比較することである．カテゴリ (b) は，カテゴリ (a) において実装して実験した方法論の元文献で報告されているスコアを示している [3]．カテゴリ (c) は，最近提案された psc と SimHash ベースの探索手法を取り入れた文献の結果を示す[*6]．最後にカテゴリ (d) は，現在のトップスコアを獲得した方法論の結果である．これらは，DQN [2]，double DQN（DDQN）[4]，Gorila [5]，Bootstrapped DQN [6]，Dueling network [7] の最も高いスコアを掲示している．これらの手法は基本的には特別な探索戦略は実装されていない．

表 3 の結果は，従来研究に合わせてスコアの平均値で評価を行っている．しかしながら平均値だけでは手法に有意な差があるのかを判断することは困難である．そこで，本

[*5] スレッド数に関しては，元文献 [8] では 16 を用いているが，本論文では，学習時間を短縮するため 56 を用いた．

[*6] DDQN+psc の詳細はスコアは元論文 [3] では報告されていないので，文献 [10] の報告値を用いた．

表 3 提案手法の結果，ベースラインおよび他の強化学習アルゴリズムとの比較．太字の数字は各ゲームの最良の結果を示す

Table 3 Results of our proposed method and comparison with baseline and current top methods. Boldface numbers indicate best result on each game.

Cat.	method	Enduro	Freeway	Gravitar	Montezuma	Private	Solaris	Venture
(a)	A3C（ベースライン：再実験）	0.0	0.0	283.3	3.3	160.1	3,287.3	0.0
	A3C+psc（再実験）	181.7	22.1	311.7	0.0	1,855.7	2,612.0	0.0
	A3C+UCB（提案手法）	28.5	23.0	406.7	126.7	**7,643.5**	**4,622.7**	163.3
(b)	A3C（ベースライン：文献 [3] 報告値）	0.0	0.0	201.3	0.2	97.4	2,102.1	0.0
	A3C+psc [3]	694.8	30.4	238.7	273.7	99.3	2,270.2	0.0
(c)	DDQN+psc（文献 [10] 報告値）	–	29.2	–	**3,439**	–	–	369
	TRPO+picel-SimHash [10]	–	31.6	468	0	–	2,897	263
	TRPO+BASS-SimHash [10]	–	28.4	604	238	–	1,201	616
	TRPO+AE-SimHash [10]	–	33.5	482	75	–	4,467	445
(d)	DQN [2]	301.8	30.3	306.7	0.0	1,788.0	–	380.0
	DDQN [4]	319.5	31.8	170.5	0.0	670.1	–	93.0
	Gorila [5]	114.9	11.7	**1,054.5**	4.2	748.6	–	**1,245.3**
	Bootstrapped DQN [6]	1,591.0	**33.9**	286.1	100.0	1,812.5	–	212.5
	Dueling network [7]	**2,258.2**	0.0	588.0	0.0	–	2,250.8	497.0

表 4 提案手法とベースラインの各ゲームにおける結果の四分位点

Table 4 The quartile of the result in each game of the proposed method and baseline.

	method	第 1 四分位点	第 2 四分位点	第 3 四分位点
Enduro	A3C	0.0	0.0	0.0
	A3C+psc	142.0	165.5	186.5
	A3C+UCB	22.5	29.5	35.5
Freeway	A3C	0.0	0.0	0.0
	A3C+psc	21.0	22.0	23.0
	A3C+UCB	22.0	23.0	24.0
Graviter	A3C	0.0	225.0	425.0
	A3C+psc	100.0	300.0	500.0
	A3C+UCB	250.0	500.0	550.0
Montezuma	A3C	0.0	0.0	0.0
	A3C+psc	0.0	0.0	0.0
	A3C+UCB	0.0	0.0	400.0
Private	A3C	−661.0	0.0	0.0
	A3C+psc	−999.0	3,983.0	4,000.5
	A3C+UCB	0.0	9,081.0	14,600.5
Solaris	A3C	1,400.0	1,920.0	3,330.0
	A3C+psc	1,240.0	1,840.0	3,100.0
	A3C+UCB	3,100.0	5,000.0	6,400.0
Venture	A3C	0.0	0.0	0.0
	A3C+psc	0.0	0.0	0.0
	A3C+UCB	0.0	100.0	250.0

論文で再実験を行ったカテゴリ (a) に関して，詳細な実験結果として四分位点を**表 4** に示す．

次に，「A3C+UCB（提案法）と A3C（ベースライン：再実験)」および「A3C+UCB（提案法）と A3C+psc（再実験)」の間で，それぞれ独立に「2 つの手法間で差がない」という帰無仮説のもと検定を行った．検定には，マン・ホイットニーの U 検定を用いた．その結果を**表 5** に示す．

表 3，表 4，表 5 のから次のことがいえる．

(1) ベースラインの A3C に関して，文献 [3] の報告値と本論文の実験結果はほぼ同じとなった．

これは本論文で用いている A3C の実装および実験設定が，文献 [3] で行われたものを再現できていることを示す 1 つの証拠になると考えられる．同様に，本論文での実験の報告値の信頼性を担保する結果となっている．

(2) A3C+UCB は一貫してベースラインの A3C より良い結果を出している．この結果から UCB ベースの探索戦略はベースラインの A3C と比較して高い報酬が得られる環境の探索に成功している．

(3) A3C+UCB（提案手法）の “Private Eye” と “Solaris” のそれぞれの平均スコアは **7643.5**，および，**4622.7** に到達した．これは比較した手法の中では最高の成績

表 5 マン・ホイットニーの U 検定を用いた有意差検定の結果：* は A3C+UCB と A3C の結果に対して有意水準 1%で有意な差が認められたゲーム．† は A3C+UCB と A3C+psc の結果に対して有意水準 1%でスコアに有意な差が認められたゲーム

Table 5 Results of significance test by Mann-Whitney U test.

method	Enduro	Freeway	Gravitar	Montezuma	Private	Solaris	Venture
A3C	*	*	*	*	*	*	*
A3C+psc	†			†	†	†	†

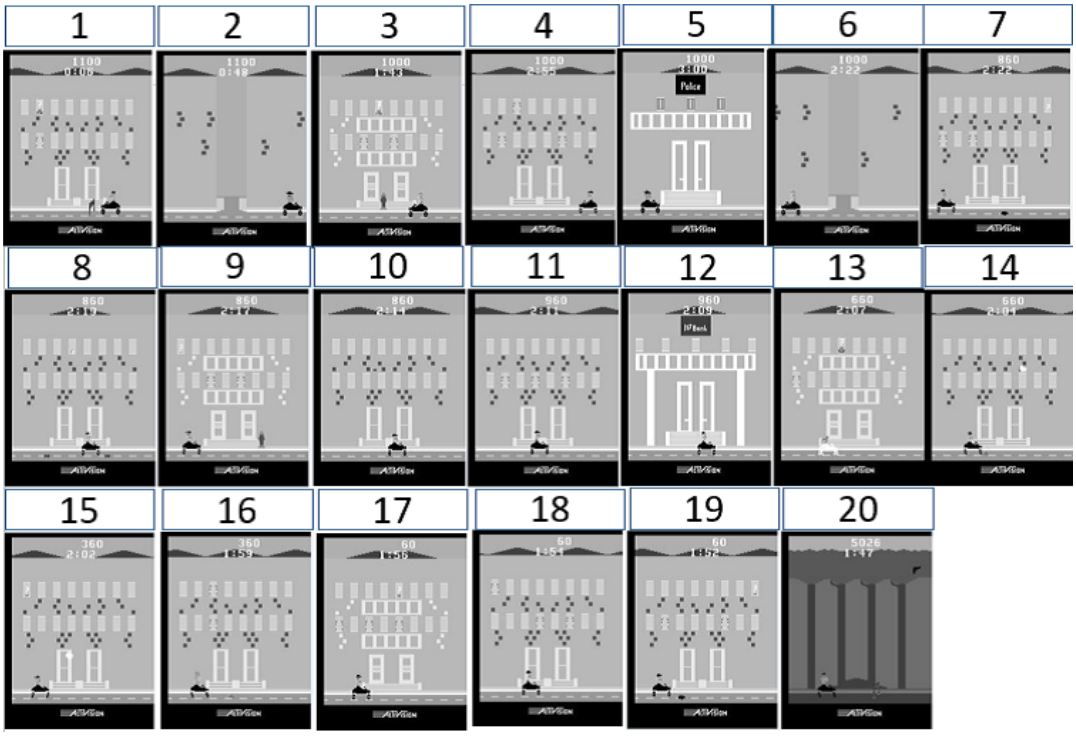


図 2 Private Eye のマップ

Fig. 2 The map of Private Eye.

であった．

(4) A3C+UCB と A3C を比較すると，7 種類すべてのゲームにおいて有意水準 1%で有意な差が認められた．

(5) A3C+UCB と A3C+psc を比較すると，“Enduro”，“Montezuma”，“Solaris”，“Venture” の 5 ゲームに関して有意水準 1%で有意な差が認められた．ただし “Enduro” に関しては A3C+psc の方が平均スコアが高いので，A3C+psc の方が有意に良い．また “Freeway”，“Gravitar” に関しては有意水準 1%で有意な差は認められなかった．

6.2 提案手法の Private Eye に対する考察

ここでは Atari 2600 の中で最も難しいゲームの 1 つである Private Eye で，提案手法である A3C+UCB により獲得した戦略を調べ，比較した手法の中で最高記録を達成した本質的な利点を考察する．このゲームでは，メインキャラクタが車を運転しながら町中にあるアイテムを集め，特定の場所に運ぶことを目的としている．アイテムを取得したりアイテムを特定の場所まで運ぶことで正の報酬（得点）が得られる．また，それ以外では，正の報酬を得ることはできない．さらにアイテムの数はとても少ないため，正の報酬を獲得できる局面は非常に限定的であり，負の報酬を獲得できる局面も同様であるため疎な報酬環境ゲームに分類されている．

Private Eye には，エージェントが高い報酬を得る戦略の獲得を妨害するいくつかの要因があると考えられる．具体的には，敵や障害物にぶつかると負の報酬（スコアが減る）となる．また，高い正の報酬はスタート地点から，遠い局面に配置されている．**図 2** に，Private Eye のマップ全体の一部を示す．このマップは 20 もの建物で構成されており，番号の 7 と 8，14 と 15 はそれぞれつながっている．メインキャラクタの初期位置は 4 番である．正の報酬を得られる場所は 3，7，9，11，13，15，17 番目の建物で相対的に小さい 100 点の報酬，そして 19 番目の建物で相対的に大きい 5,000 点の報酬が得られる．敵などは 7，8，10，13，14，15，16，19 番目の建物に存在する．

このゲームにおいて高得点を獲得するには，一番奥の建物までキャラクタを動かし，大きい報酬を得る必要がある

表 7 各提案モジュール個別の効果検証
Table 7 The effects of each proposed module.

method	Enduro	Freeway	Gravitar	Montezuma	Private	Solaris	Venture
提案手法（単一の戦略）	32.0	22.1	313.0	0.0	1,409.0	170.0	0.0
提案手法 w/ psc	4.0	19.6	100.0	20.0	967.0	2,798.0	0.0
提案手法	28.5	24.0	406.7	126.7	7,643.5	4,622.7	163.3

表 6 Private Eye の画面 19 まで到達した回数
Table 6 Number of times agents reached screen 19.

method	到達回数
A3C	1
A3C+psc	8
A3C+UCB	17

が，そのためには負の報酬を得ても遠くまで探索を進める戦略が必須となる．従来法では，学習が進むに連れて，エージェントが負の報酬を獲得しないようにこれらの局面を回避し，初期位置近くの小さな報酬を獲得できる局面を重点に探索する傾向があると容易に推測できる．つまり，大きな正の報酬が得られる局面に到達するためには，長期的な探索戦略を持っている必要がある．提案手法は，初期位置近くの小さい報酬を獲得しても，探索報酬によって，未知の局面であるより奥の建物にエージェントを進ませようとする．

具体的な検証として，それぞれの手法のそれぞれの試行で，画面 19 まで到達できた回数をカウントした．その結果を**表 6** に示す．この結果からも分かるように，提案法では 30 回のうち半数以上の 17 回は高い報酬がある建物の場所まで到達できた．一方，従来法は 1 回のみであった．

結果として，提案手法においてエージェントは未知の局面を有効に探索し，大きな報酬を得ることに成功したと考えられる．このように，報酬が得られる状態が非常に少ないゲームにおいても，提案手法の探索が有効に機能していることが分かった．

6.3 各提案モジュールの効果

提案手法の各モジュールの効果を検証するために次の設定で実験を行った．

- 提案手法（単一の戦略）：戦略 π' を用いず，従来法で用いられるように，戦略 π の更新に探索報酬 e を加算して使用．
- 提案手法 w/ psc：式 (8) を式 (5) で置き換え

表 7 に結果を示す．なお，提案手法の結果は，表 3 で示した結果を再掲したものである．

まず，提案手法（単一の戦略）と提案手法の結果を比べることで，2 種類の戦略を別に学習することの効果を見ることができる．この結果から，明らかに 2 種類の戦略を用いることの効果が確認できる．

また，提案手法 w/ psc と提案手法の結果を比較することで，UCB に基づく探索基準を用いることの効果を見ることができる．psc を用いることでも一定の効果があるが，UCB に基づく探索基準の方が効果が高いことが確認できる．

このように，提案手法は複数のモジュールを複合した結果としてうまく効果を発揮していることがみてとれる．

7. まとめ

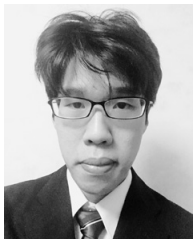
本論文では，報酬が疎な環境に対する深層強化学習に関して議論を行い，報酬が疎な環境に適した深層強化学習の枠組みを 1 つ提案した．提案手法の効果を検証するため，深層強化学習法のベンチマークとしてよく用いられている Arcade Learning Environment（ALE）に実装されている Atari 2600 のゲームの中で，疎な報酬環境ゲームに分類されるゲームを用いて実験を行った．提案法は，ベースラインとなる A3C と比べて疎な報酬環境ゲームのスコアを大きく向上できることを示した．また，疎な報酬環境ゲームの中で，“Private Eye” と “Solaris” においては比較した手法の中で最高の成績であった．

提案法の活用戦略の行動は，探索戦略という異なる戦略からサンプリングされたエピソードを利用しているため，活用戦略の方策勾配は探索戦略によるバイアスがかかる．つまり提案手法はヒューリスティックな方法論であり，活用戦略の理論的な最適性の保証を与えられるかは現状不明である．学習の収束性や理論的な解析に関しては，今後の課題として取り組みたいと思っている．

参考文献

[1] Bellemare, M.G., Naddaf, Y., Veness, J. and Bowling, M.: The Arcade Learning Environment: An evaluation platform for general agents, *Journal of Artificial Intelligence Research*, Vol.47, pp.253–279 (2013).
[2] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning, *Nature*, Vol.518, No.7540, pp.529–533 (2015).
[3] Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D. and Munos, R.: Unifying count-based exploration and intrinsic motivation, *Advances in Neural Information Processing Systems 29*, pp.1471–1479 (2016).
[4] Van Hasselt, H., Guez, A. and Silver, D.: Deep Reinforcement Learning with Double Q-Learning, *Proc. 30th AAAI Conference on Artificial Intelligence*, pp.2094–

2100 (2016).
[5] Nair, A., Srinivasan, P., Blackwell, S., Alcicek, C., Fearon, R., De Maria, A., Panneershelvam, V., Suleyman, M., Beattie, C., Petersen, S., et al.: Massively parallel methods for deep reinforcement learning, *ICML Deep Learning Workshop* (2015).
[6] Osband, I., Blundell, C., Pritzel, A. and Van Roy, B.: Deep exploration via bootstrapped DQN, *Advances In Neural Information Processing Systems 29*, pp.4026–4034 (2016).
[7] Ziyu, W., Nando de, F. and Marc, L.: Dueling Network Architectures for Deep Reinforcement Learning, *Proc. 33rd International Conference on Machine Learning*, pp.1995–2003 (2016).
[8] Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T.P., Harley, T., Silver, D. and Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning, *Proc. 33rd International Conference on Machine Learning* (2016).
[9] Strehl, A.L. and Littman, M.L.: An analysis of model-based interval estimation for Markov decision processes, *Journal of Computer and System Sciences*, Vol.74, No.8, pp.1309–1331 (2008).
[10] Tang, H., Houthooft, R., Foote, D., Stooke, A., Chen, X., Duan, Y., Schulman, J., De Turck, F. and Abbeel, P.: #Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning, arXiv preprint arXiv:1611.04717 (2016).
[11] Pathak, D., Agrawal, P., Efros, A.A. and Darrell, T.: Curiosity-driven exploration by self-supervised prediction, *Proc. 34th International Conference on Machine Learning*, pp.2778–2787 (2017).
[12] Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O. and Zaremba, W.: Hindsight Experience Replay, *Advances in Neural Information Processing Systems 30*, pp.5048–5058 (2017).
[13] Riedmiller, M.A., Hafner, R., Lampe, T., Neunert, M., Degrave, J., Van de Wiele, T., Mnih, V., Heess, N. and Springenberg, J.T.: Learning by Playing - Solving Sparse Reward Tasks from Scratch, arXiv preprint arXiv:1802.10567 (2018).
[14] Lai, T.L. and Robbins, H.: Asymptotically efficient adaptive allocation rules, *Advances in Applied Mathematics*, Vol.6, No.1, pp.4–22 (1985).
[15] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M.: Playing Atari With Deep Reinforcement Learning, *NIPS Deep Learning Workshop* (2013).
[16] Nair, V. and Hinton, G.E.: Rectified linear units improve restricted boltzmann machines, *Proc. 27th International Conference on Machine Learning*, pp.807–814 (2010).
[17] Andoni, A. and Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, *47th Annual IEEE Symposium on Foundations of Computer Science*, pp.459–468 (2006).



水上 直紀 （正会員）

2015 年東京大学大学院工学系研究科修士修了．2018 年東京大学大学院工学系研究科博士課程単位取得退学．同年 HEROZ 株式会社入社．ゲーム AI に関する研究に従事．

鈴木 潤 （正会員）

1999 年慶應義塾大学理工学部数理科学科卒業．2001 年同大学院理工学研究科計算機科学専攻修士課程修了．同年日本電信電話株式会社入社．2005 年奈良先端大学院大学博士後期課程修了．2008～2009 年 MIT CSAIL 客員研究員．2018 年より東北大学大学院情報科学研究科に所属．博士（工学）．主として自然言語処理，機械学習に関する研究に従事．ACL，言語処理学会各会員．

亀甲 博貴 （正会員）

2018 年東京大学大学院工学系研究科博士課程修了．博士（工学）．同年より京都大学学術情報メディアセンター助教．自然言語処理，ゲーム AI 等に関する研究に従事．言語処理学会会員．

鶴岡 慶雅 （正会員）

2002 年東京大学大学院工学系研究科電子工学専攻博士課程修了．博士（工学）．科学技術振興事業団研究員，マンチェスター大学 Research Associate，北陸先端科学技術大学院大学准教授，東京大学大学院工学研究科准教授を経て，2018 年より東京大学大学院情報理工学系研究科教授，現在に至る．自然言語処理およびゲーム AI の研究に従事．