情報処理学会論文誌　数理モデル化と応用　Vol.14　No.1　1–11 (Jan. 2021)

# 深層強化学習における時系列的内部報酬生成器による探索の改善

村上 知優[1,†1,a)]　森山 甲一[1]　松井 藤五郎[2]　武藤 敦子[1]　犬塚 信博[1]

受付日 2020年1月31日，再受付日 2020年3月24日，
採録日 2020年7月21日

**概要**：近年，高次元状態における強化学習手法として深層強化学習という手法が注目されている．しかし，深層強化学習を含む強化学習全般において，報酬が疎な環境における学習が困難であることが知られている．この問題を解決する手段として，目新しい状態の訪問に対して内的な報酬を発生させ，エージェントに多様な状態への訪問を促進させる手法が存在する．本研究ではそれを時系列的なものへ拡張し，目新しい状態遷移に対して内部報酬を生成するようにした．これにより部分観測マルコフ決定過程における探索にも対応できるようにし，実験を行った結果，その有効性を確認した．

キーワード：強化学習，深層学習，深層強化学習，探索，内部報酬

# Exploration Improvement by Sequential Intrinsic Reward Generator in Deep Reinforcement Learning

Kazuhiro Murakami[1,†1,a)]　Koichi Moriyama[1]　Tohgoroh Matsui[2]　Atsuko Mutoh[1]
Nobuhiro Inuzuka[1]

Received: January 31, 2020, Revised: March 24, 2020,
Accepted: July 21, 2020

***Abstract:*** Deep reinforcement learning is working well in the environment with high dimensional states. However, it is difficult for a reinforcement learning agent to learn an optimal policy in the environment where it hardly obtain rewards. Curiosity-driven exploration is a solution that gives intrinsic rewards to the agent in unfamiliar states to encourage it for visiting various states. This work proposes Sequential Intrinsic Reward Generator (SRG), which extends curiosity-driven exploration to a sequence of states and gives the agent intrinsic rewards for unfamiliar state transitions. Due to this sequential property, SRG is promising to work well also in partially observable Markov decision processes. The result of experiments shows that SRG worked better than other methods in such environments.

***Keywords:*** reinforcement learning, deep learning, deep reinforcement learning, exploration, intrinsic reward

## 1. はじめに

近年，様々な場面で物事の自動化が進められている．従来は人間が動作ロジックをすべて定義することで行われてきたが，現実世界のより複雑なものを自動化するにあたって，動作ロジックをすべて定義することは非常に難しく，現実的でない．こうした背景のもと，動作ロジック自体を機械が獲得する機械学習の分野が注目されており，なかでも与えられた環境下で最適な行動を学習する強化学習が大

[1] 名古屋工業大学大学院工学研究科情報工学専攻
Department of Computer Science, Graduate School of Engineering, Nagoya Institute of Technology, Nagoya, Aichi 466–8555, Japan
[2] 中部大学生命健康科学部臨床工学科
Department of Clinical Engineering, College of Life and Health Sciences, Chubu University, Kasugai, Aichi 487–8501, Japan
[†1] 現在，フューチャー株式会社
Presently with Future Corporation
[a)] k.murakami.638@nitech.jp

きな成果をあげている [1].

また，近年画像認識や自然言語処理，音声認識などの分野で深層学習が大きな成果をあげている．深層学習は入力データから特徴を抽出することに長けており，人間には認識できないような特徴を抽出することができる可能性を秘めている．実際，画像認識のコンペティションとして有名な ImageNet Large Scale Visual Recognition Challenge（ILSVRC）[2] において，ResNet [3] が人間の誤認率 5.1%を下回る誤認率 3.6%を記録し，話題となった．

さらに，深層学習と強化学習を組み合わせた深層強化学習の手法として，Deep Q Network（DQN）[4], [5] が提案された．DQN は Atari2600 [6] のゲーム画面を状態とし，得られたスコアを報酬とすることで，上級者を上回るスコアを出せる方策を獲得するに至っている．また，方策の単調改善を目的に Proximal Policy Optimization（PPO）[7] が提案され，後の研究で幅広く用いられている．

一方で，Atari2600 には人間よりもスコアの劣るゲームも存在し，主に次の要素が存在するゲームがそれにあたる．

(1) 過去の情報を覚えておかなければならないようなゲーム．
(2) あらゆる状態の中で報酬がごく一部の状態でしか得られないゲーム．

(1) のゲームに関しては，行動を決定する際に過去の情報を参照する機能を追加することで解決が図られている [8]．また，強化学習では人間が設計した報酬を可能な限り多く集めるためにはどのように行動すればよいかを学習するが，どの状態で報酬が獲得できるかという知識があらかじめ存在しないため，まずは様々な状態を訪問して報酬を獲得しなければならない．しかし，(2) のようなゲームでは報酬を見つけることがきわめて困難であるため，深層強化学習を含む強化学習全般において解くことが難しい．実際に DQN や PPO では Atari2600 のそのようなゲームで，ほとんどスコアを獲得できないことが報告されている [5], [7]．

この問題の解決策として，人間が設計した報酬を可能な限り多く集めるための行動を学習しつつも，報酬を探すための行動を学習するという手法が存在し，そのなかでも未知の状態に遭遇した際に報酬を発生させる手法を好奇心探索 [9] という．好奇心探索では人間が設計した報酬とは別にエージェントが自発的に報酬を生成する機構をエージェントに実装する．この機構はエージェントがまだ訪問したことのない状態を訪問したときに報酬を発生するものになっているため，エージェントは未知の状態を求めて行動を決定するようになる．一方で従来手法はある状態の目新しさを評価することで報酬を生成しているため，マルコフ決定過程（MDP）における探索にのみ対応している．また，行動の目新しさについては考慮されていない．そこで本研究ではある状態のみの目新しさを測るのではなく，状態の系列に対する目新しさも測るよう拡張し，同時に目新しい行動についても評価できるよう改良した．これにより部分観測マルコフ決定過程（POMDP）における探索にも対応することができる．このような状態の評価のみでは十分に報酬を探すことができない場合にも対応するような新たな機構を提案することで，好奇心探索の性能向上を図る．

## 2. 好奇心探索

好奇心探索はエージェントに人間でいう好奇心にあたるものを搭載することで未知の状態への訪問を促進させ，報酬を探させるというものである．未知の状態への訪問に対する報酬（内部報酬）を与えることで，未知の状態を探す方策を学習することになる．それと同時に人間の設計した本来の報酬（外部報酬）を獲得することで，本来エージェントに学習してほしい方策も学習することができる．したがって，探索するための方策と本来の方策の混合方策を学習することになる．しかし，最終的に獲得してほしいのは本来の方策であるため，好奇心探索では既訪問状態に対する内部報酬がしっかりと減少し，最終的には 0 に収束するような内部報酬生成器をエージェントに組み込むことが重要である．

### 2.1 深層強化学習における内部報酬生成器

好奇心探索を実現するためには訪問済みの状態を記録する必要がある．しかし，深層強化学習は状態の次元数が大きいことを仮定しているので，これらをすべて記録することは難しい．よって一般的に次のような枠組みが用いられる．

$$\boldsymbol{y} = f(\boldsymbol{s}) \tag{1}$$

$$i = ||\boldsymbol{t} - \boldsymbol{y}||_2^2 \tag{2}$$

ここで，$\boldsymbol{s}$ は状態，$f$ は Deep Neural Network（DNN），$\boldsymbol{y}$ はその出力，$\boldsymbol{t}$ は $\boldsymbol{y}$ に対する教師データ，$i$ は内部報酬を表す．このように DNN によって状態から何らかの推定を行い，その誤差を用いて内部報酬を計算する．この枠組みの挙動を考えてみると，頻度の大きい入力に対する出力は妥当なものとなり，誤差が小さくなって内部報酬も小さくなる．逆に頻度の小さい入力に対する出力は妥当性に欠けるため，誤差が大きくなって内部報酬も大きくなる．したがって，見慣れた状態に対する内部報酬は小さくなり，見慣れない状態に対する内部報酬は大きくなる．ではこの DNN に何の推定をさせるかが問題となるが，このときに好奇心探索の項であげた，「既訪問状態に対する内部報酬がしっかりと減少する」という条件が重要となってくる．この条件を満たすにあたって，内部報酬がどのようなときに大きくなるのかを考えると，以下の 4 つの要因に分けられる [10]．

(1) 入力される頻度の少ない状態が入力された．
(2) 入力に対する教師データが一意でなく，label corruption

が起こった．
(3) 入力された情報からでは推定ができず，under fitting が起こった．
(4) 過学習が発生した．

要因 (1) は好奇心探索を実現するうえで不可欠な要素であると同時に，これ以外の要因で内部報酬が大きくなることは好ましくない．要因 (2) はこれによって誤差が一向に減らなくなってしまい，結果的にどの状態を訪問しても内部報酬を獲得できてしまう．要因 (3) はそもそもその推定タスクを解くことができず，誤差が一向に減らないことで，どの状態においても内部報酬が大きいまま維持されてしまう．要因 (4) は DNN の重みが局所最適解に陥り，それ以上誤差が減少しなくなることで内部報酬が減少しないという状況である．要因 (4) は DNN の構造や学習則に依存することであるが，要因 (2) と要因 (3) は何の推定を行うかに依存することであるため，これらを回避するような推定タスクにする必要がある．

### 2.2 Random Network Distillation

Random Network Distillation（RND）[10] は，上記要因 (2)，(3) の回避を実現した内部報酬生成器であり，これらの回避により Atari2600 のゲームの 1 つである Montezuma Revenge という環境で初めて人間のスコアを超えることができた手法である．RND では方策の学習手法に PPO を用い，内部報酬生成器に 2 つの DNN を用いる．そして内部報酬の計算は次のように行う．

$$\boldsymbol{y}_1 = \hat{f}_1(\boldsymbol{s}) \tag{3}$$

$$\boldsymbol{t}_1 = f_1(\boldsymbol{s}) \tag{4}$$

$$i = \frac{1}{N_1}||\boldsymbol{t}_1 - \boldsymbol{y}_1||_2^2 \tag{5}$$

ここで，$N_1$ は $\boldsymbol{t}_1$ および $\boldsymbol{y}_1$ の次元数である．また，$\hat{f}_1$ は学習を行う DNN であるが，$f_1$ は学習を行わない DNN である．$\hat{f}_1$ を Predictor Network，$f_1$ を Target Network という．つまり，Target Network の出力を教師データとして Predictor Network の学習を行うということである．これにより，入力に対する教師データの一貫性が保たれるため，要因 (2) の回避が可能となる．また，要因 (3) は Target Network と Predictor Network の入力を同じにすることで回避することができている．

## 3. 提案手法

本研究ではある状態のみの目新しさを測るのではなく，状態の系列に対する目新しさも測るように拡張した「Sequential Intrinsic Reward Generator（SRG）」を提案する．

### 3.1 深層強化学習とリカレント層

深層強化学習ではリカレント層を用いて状態や行動の履歴を内部状態という形で保持することで，POMDP に対応できることが知られている．実際に深層強化学習で Long Short Term Memory（LSTM）[11] と呼ばれるリカレント層を用いることで，POMDP における有効性を確認した例が存在する [12]．SRG ではこのリカレント層による効果を探索へ組み込む．

### 3.2 時系列拡張による本質的な違い

目新しさを測る対象をタプルで表現すると，RND と SRG は次のように記述できる．SRG ではこのタプル自体の目新しさを評価する．

$$RND : \{\boldsymbol{s_t}\}$$

$$SRG : \{\boldsymbol{s_{t-l}}, \boldsymbol{s_{t-l+1}}, \cdots \boldsymbol{s_t}\}$$

ここで，$l$ は目新しさを評価する系列長である．MDP では行動をとることで状態が遷移することが仮定されているので，SRG のタプルに含まれる状態と次の状態の間には暗に行動が含まれていることになる．よって，目新しい行動についても考慮することができる．

状態のみの評価と系列の評価の違いを**図 1** を用いて具体的に考えてみる．緑色の長方形は状態を表し，黒色のエッジは両端の状態を相互に行き来することができることを意味する．はじめエージェントは $s_0$ におり，他の状態は未訪問であるとする．RND は $s_0 \to s_1 \to s_2 \to s_3 \to s_0$ と遷移すると，$s_3$ を訪問したときまでは内部報酬が発生するが，最後に再び $s_0$ を訪問したときは内部報酬が発生しない．これはすでにエージェントが $s_0$ を訪問済みであるからである．それに対して SRG は一番最初のタプルが $\{s_0\}$ であり，最後のタプルは $\{s_0, s_1, s_2, s_3, s_0\}$ であるため，タプルの内容が異なり，目新しい系列であるといえる．よって一番最後に $s_0$ を訪問した際にも内部報酬が発生する．したがって，RND は目新しい状態の訪問に対して内部報酬を生成するのに対し，SRG は目新しい状態遷移に対して内部報酬を生成する．また，SRG は過去の状態や行動の履歴を保持し，その目新しさを測っているため，POMDP の探索に対応することができる．

### 3.3 Sequential Intrinsic Reward Generator

SRG では上記タプルに対する目新しさを測るために LSTM を用いる．RND の内部報酬生成器に新たに LSTM を含む DNN を追加し，次のように内部報酬を計算する．

$$\boldsymbol{y_1} = \hat{f}_1(\boldsymbol{s}) \tag{6}$$

$$\boldsymbol{t_1} = f_1(\boldsymbol{s}) \tag{7}$$

$$\boldsymbol{y_2} = \hat{f}_2(\phi(\boldsymbol{s})) \tag{8}$$

$$\boldsymbol{t_2} = f_2(\phi(\boldsymbol{s})) \tag{9}$$

$$i = \alpha\frac{1}{N_1}||\boldsymbol{t_1} - \boldsymbol{y_1}||_2^2 + (1-\alpha)\frac{1}{N_2}||\boldsymbol{t_2} - \boldsymbol{y_2}||_2^2 \tag{10}$$

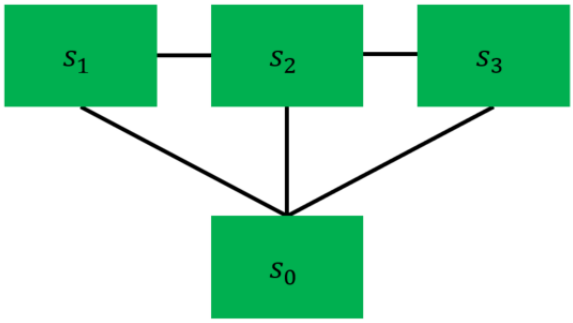



図 1　時系列評価の意義を示す環境の例
**Fig. 1**　An environment showing importance of sequence based evaluation.

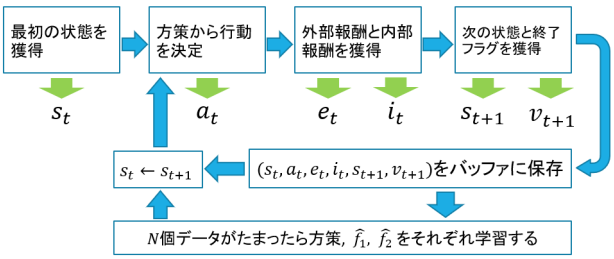



図 2　SRG のアルゴリズム
**Fig. 2**　SRG algorithm.

ここで，$\alpha$ は状態のみの評価と時系列の評価の内分比である．$N_1$，$N_2$ はそれぞれ $\boldsymbol{t_1}$，$\boldsymbol{y_1}$ と $\boldsymbol{t_2}$，$\boldsymbol{y_2}$ の次元数を表す．また，$\hat{f}_2$，$f_2$ が新たに追加した DNN で，$f_2$ にはリカレント層が含まれていないが，$\hat{f}_2$ に LSTM が含まれている．なお，RND と同じように $f_1$，$f_2$ は学習を行わず，$\hat{f}_1$，$\hat{f}_2$ は学習を行う．LSTM には内部状態として過去に入力された状態が保持されているため，今獲得した状態を入力することで上記タプルに対する出力を行うことができる．さらに，LSTM は内部状態の更新を今入力された情報を加算することで行うため，無限長の系列を扱うことができる．そして内部状態は任意のタイミングでリセット可能である．よって，先のタプルに存在する $l$ は自動的に無限となるが，エピソードをまたぐと過去のエピソードの系列が影響してしまうため，エピソードの終了時に内部状態をリセットする．また，SRG はタプルを時間方向へ拡張するため，DNN の学習が難しくなり，RND と比べて内部報酬が減少しにくくなる．このことは前章で説明した要因 (4) による内部報酬が大きいまま維持されることを引き起こす可能性がある．よってこれを防ぐために $N_1 > N_2$ とし，状態を次元圧縮を行う関数 $\phi$ によって十分学習可能な次元数まで圧縮したうえで $\hat{f}_2$，$f_2$ に入力し，学習を行う．これにより入力の多様性がある程度失われるため，$\hat{f}_2$ の学習がしやすくなる．以上をふまえて SRG のアルゴリズムは**図 2** のようになる．図中の $a$ は行動，$e$ は外部報酬，$i$ は内部報酬，$v$ はエピソードの終端かどうかを表すフラグである．

表 1　各環境の POMDP 性
**Table 1**　The degree of POMDP for each environment.

| 環境名 | POMDP 性 |
|---|---|
| Pong（報酬設定変更） | 高 |
| Gravitar | 低 |
| Montezuma Revenge | 高 |
| Pitfall | 高 |
| Private Eye | 高 |
| Solaris | 低 |
| Venture | 低 |

## 4.　実験

エージェントが方策を学習する環境として，OpenAI Gym [6] に存在する Atari2600 を用いた．Atari2600 には 50 個以上の様々なゲームが存在するが，DQN や PPO などの基本的な深層強化学習手法によって，人間のスコアを越えることができていないゲームの中でも特に 1 章で述べた (2) の性質を持つゲームに着目して実験を行う．具体的には，文献 [10] で着目されている報酬の疎なゲーム（Gravitar，Montezuma Revenge，Pitfall，Private Eye，Solaris，Venture）および，報酬が疎になるよう設定を変更した Pong というゲームにおける，PPO，RND と SRG の差異について検証する．エージェントが得ることのできる状態は，ゲーム画面を画像として取得したものである．各環境の説明，および各設定の説明を以下に示す．また，**表 1** は各環境の POMDP 性の高低をまとめたものである．

本実験では CPU に Intel Xeon E5-2650 v4 を 2 基，GPU に NVIDIA GeForce GTX 1080 Ti を 2 基用い，計算ライブラリとして Chainer，ChainerRL，CUDA，cuDNN を用いた．

### 4.1　各ゲームの説明

#### 4.1.1　Pong

Pong は卓球を模したゲームで，画面端のバーを操作して球を跳ね返し，相手のバーよりも奥に球を到達させることで +1 点のスコアが得られるゲームである．球を打ち合い，先に 21 点に達したプレイヤが勝利となり，エピソード終了である．エージェントはスコアを獲得したときに正の外部報酬を獲得し，逆に相手に点を取られた場合は負の報酬が与えられる．Pong は本来外部報酬が疎な環境ではないが，これをゲームに勝利したときのみ +1 の外部報酬を与えるようにすることで，外部報酬が疎な環境へ変更した．この環境での実験はできるだけ早くゲームに勝利するような強い方策を学習することが目的ではなく，SRG の実際の挙動を確認することが目的となる．

#### 4.1.2　Gravitar

Gravitar は宇宙船から弾を発射して惑星上に存在する対空砲を破壊することでスコアを獲得するゲームである．対

空砲の弾に当たるもしくは惑星や壁などの障害物に接触すると残機が減り，残機が 0 になった時点でエピソード終了となる．エージェントは宇宙船を操作し，対空砲を破壊すると正の外部報酬が得られる．

この環境は POMDP 性は低いが，エージェントが操作する宇宙船に強力な慣性が存在するゲームとなっている．たとえば右に移動しているときに左に方向転換したい場合，左移動を選択しても右成分の慣性がなくなるまで実際に左に移動し始めない．そしてこの慣性は速度が大きいほど大きくなるため，物体が高速で移動しているほど方向転換が完了するまでに多くの時間がかかる．このように物体の速度に応じた行動選択を求められる環境である．SRG では時系列と目新しい行動について評価を行うため，今の宇宙船の速度を考慮したうえで適切な行動のタイミングを発見しやすいと考えられる．

#### 4.1.3　Montezuma Revenge

Montezuma Revenge はキャラクタを操作して様々な部屋と障害物によって構成される迷路から脱出するゲームである．途中扉を開けるための鍵や，敵を倒すことができるアイテムなどが存在し，それらを使って道を開拓しつつゴールを探す．エージェントはキャラクタを操作し，特定の部屋に存在するアイテムを獲得する，鍵のかかった扉を開ける，アイテムを使って敵を倒すことでスコアが得られる．キャラクタが高所から落下したり敵や障害物に接触すると残機が減り，残機が 0 になった時点でエピソード終了となる．エージェントはスコアを獲得したときに正の外部報酬を得ることができる．

この環境は外部報酬が非常に疎な環境として知られており，なかには一定間隔で出現と消滅を繰り返す障害物があるような POMDP 性の高い部屋も存在する．Montezuma Revenge では基本的に部屋の一番奥に 1 つのアイテムが存在し，それを獲得することでその部屋はほぼ探索終了となる構造になっている．障害物のなかには 1 度触れると消滅するものも存在するため，次に説明する Pitfall よりは広範な探索がしやすい環境となっている．POMDP 性の高い部屋や，扉を開けるためにまず鍵を獲得しなければならないといった順序関係の存在から，SRG の特性が有効であると考えられる．

#### 4.1.4　Pitfall

Pitfall はキャラクタを操作して様々な部屋を訪問しながら迫り来る障害物を回避しつつ制限時間内に可能な限りスコアを伸ばすゲームである．エージェントはキャラクタを操作し，特定の部屋に落ちているアイテムを獲得することでスコアを獲得することができる．逆に障害物に接触すると残機が減ると同時にスコアが減少する．残機が 0 になる，または残り時間が 0 になるとエピソード終了となる．エージェントはスコアを獲得すると正の外部報酬が得られ，スコアが減少すると負の外部報酬が与えられる．

この環境は障害物の回避が Montezuma Revenge と比べてとても難しいものになっており，なかには一定の時間間隔で出現と消滅を繰り返す落とし穴など，POMDP 性が高いものも存在する．さらにアイテムが落ちている部屋が非常に限られており，Montezuma Revenge よりも正の外部報酬が疎な環境となっているため，とても高度な探索能力が要求される．SRG による POMDP への対処が有効性を示すと考えられる．

#### 4.1.5　Private Eye

Private Eye はキャラクタを操作して制限時間内に様々な部屋に存在するアイテムを回収し，それを特定の部屋まで運ぶゲームである．アイテムを回収して運ぶというタスクを繰り返し，最後のアイテムを運び終わった時点でゲームクリアとなる．アイテムを回収したときとそれを特定の部屋に運んだ時にスコアが得られる．またゲーム内には様々な障害物が存在し，接触することでスコアが減少する．ゲームをクリアする，または残り時間が 0 になるとエピソード終了となる．エージェントはスコア獲得時に正の外部報酬を獲得し，スコア減少時に負の外部報酬が与えられる．

この環境は訪問できる部屋と回収できるアイテムに順序関係が存在するため POMDP 性が高い．順序関係が重要なため，元来た部屋に戻らなければならないという状況が多分に存在する環境となっている．SRG では順序関係の目新しさを考慮した探索を行うことができるため，アイテムの回収に成功しやすいと考えられる．

#### 4.1.6　Solaris

Solaris は宇宙船を操作して画面奥から迫り来る敵を倒し，スコアを伸ばすゲームである．エージェントは宇宙船を操作し，弾を発射して敵を倒すことでスコアを獲得することができる．障害物に接触すると残機が減り，残機が 0 になるとエピソード終了となる．エージェントはスコアを獲得することで正の外部報酬を得られる．

この環境は POMDP 性が低く順序関係も特に存在しないため，RND と SRG に特に差異は見られないと考えられる．

#### 4.1.7　Venture

Venture はキャラクタを操作して複数ある部屋から任意の部屋に入り，そのなかの敵を倒すことでスコアを伸ばすゲームである．エージェントはキャラクタを操作し，弾を発射して敵を倒すことでスコアを獲得することができる．敵に接触すると残機が減り，残機が 0 になるとエピソード終了となる．エージェントはスコアを獲得することで正の外部報酬が得られる．

Venture も Solaris と同じように POMDP 性が低く，順序関係も特に存在しないゲームであるため，RND と SRG に特に差異は見られないと考えられる．

### 4.2　各設定の説明

#### 4.2.1　DNN の構造

PPO と RND の DNN は，それぞれ文献 [7]，[10] と同一のものを用いた．しかし，RND では方策を出力する DNN である Policy Network に PPO と異なるものを用いている．そして SRG における DNN の構造の内，Policy Network と RND の内部報酬生成部分は文献 [10] と同一である．これをふまえて SRG における DNN の構造を**図 3**，**図 4** に示す．図 3 は RND と SRG で用いられている Policy Network の構造であり，図 4 の $f_1$，$\hat{f}_1$，$f_2$，$\hat{f}_2$ は内部報酬生成器の DNN である．RND との比較を行うため，$f_1$，$\hat{f}_1$ の構造は RND と同じものを用い，新たに追加した $f_2$，$\hat{f}_2$ は好奇心探索における重要な条件である「既訪問状態に対する内部報酬がしっかりと減少する」を確実に満たすように，次のような予備実験で構造を決定した．

(1) $\hat{f}_2$ の全結合層を増やす．
(2) Montezuma Revenge でプログラムを動かす．
(3) 内部報酬の推移を確認．
(4) 初めの 100,000 ステップにおける内部報酬の減少が前回の構造よりも速ければ 1 へ戻り，遅ければ前回の構造を採用する．

この予備実験で決定した構造を用いる上で，要因 (4) による内部報酬の増大にも十分に配慮する必要があるが，本研究では実験中にそのような様子は確認できなかった．しかし，要因 (4) による内部報酬の増大が確認できた場合は，深層学習の分野でよく用いられるドロップアウト [13] などの工夫が必要になると考えられる．

#### 4.2.2　内部報酬の計算

内部報酬の計算を次の式で行う．

$$\boldsymbol{y_1} = \hat{f}_1(\boldsymbol{s}) \tag{11}$$

$$\boldsymbol{t_1} = f_1(\boldsymbol{s}) \tag{12}$$

$$\boldsymbol{y_2} = \hat{f}_2(\boldsymbol{t_1}) \tag{13}$$

$$\boldsymbol{t_2} = f_2(\boldsymbol{t_1}) \tag{14}$$

$$i = \alpha \frac{1}{N_1} ||\boldsymbol{t_1} - \boldsymbol{y_1}||_2^2 + (1-\alpha)\frac{1}{N_2}||\boldsymbol{t_2} - \boldsymbol{y_2}||_2^2 \tag{15}$$

状態の次元圧縮関数 $\phi$ として $f_1$ を用いている．$f_1$ の出力が十分次元圧縮されたものであることが前提となるが，このようにすることで $\phi$ の計算を省略することができ，アルゴリズム全体を高速化することができる．

#### 4.2.3　$\hat{f}_1$，$\hat{f}_2$ の誤差関数

誤差関数を次のように定義する．

$$E_{\hat{f}_1} = \frac{1}{M}\sum_{j=0}^{M} ||\boldsymbol{t_{1j}} - \boldsymbol{y_{1j}}||_2^2 \tag{16}$$

$$E_{\hat{f}_2} = \frac{1}{M}\sum_{j=0}^{M} ||\boldsymbol{t_{2j}} - \boldsymbol{y_{2j}}||_2^2 \tag{17}$$

$M$ は一度の重みの更新に用いるデータ数である．

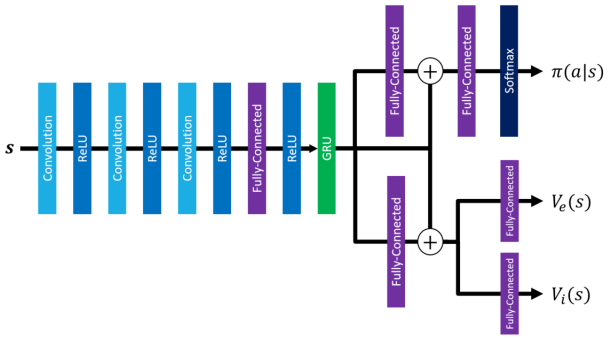



図 3　Policy Network の構造
**Fig. 3**　Policy Network architecture.

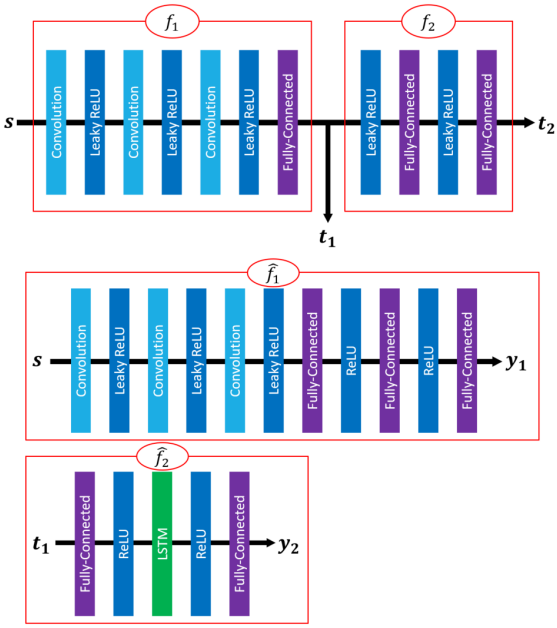



図 4　$f_1$，$f_2$，$\hat{f}_1$，$\hat{f}_2$ の構造
**Fig. 4**　$f_1$, $f_2$, $\hat{f}_1$ and $\hat{f}_2$ architectures.

表 2　ハイパーパラメータ
**Table 2**　Hyper parameters.

| ハイパーパラメータ | 値 |
|---|---|
| 並列環境数 | 32 |
| $M$ | 1,024 |
| $\alpha$ | 0.5 |
| $a$ | 0.2 |
| 外部報酬の割引率 | 0.999 |
| 内部報酬の割引率 | 0.99 |
| Policy Network の学習率 | 0.0001 |
| $\hat{f}_1$ の学習率 | 0.0001 |
| $\hat{f}_2$ の学習率 | 0.0001 |

#### 4.2.4　ハイパーパラメータ

本実験におけるハイパーパラメータの設定を**表 2** に示す．ハイパーパラメータは基本的に RND と同じものを用いた．また，$\alpha$ は 0 にすると完全に提案部分のみで内部報酬を生成することになる．この場合得られた状態はすべて既知だとしても，観測した順序が違うことで内部報酬が得

られるため，目新しい状態になかなか行かなくなるという状況が起こりうる．十分な時間学習することで最終的には順序が違うことに対する内部報酬も減少し，多様な状態を訪問することができるようになるが，非常に時間がかかるため効率が悪い．これを防ぐために $\alpha$ は 0.5 とした．しかし，Pong はほかの環境に比べて状態数が少ないため先の問題が起こりにくいと考え，$\alpha$ は 0 として実験を行った．

## 5. 結果と考察

### 5.1 Pong

エージェントは初め球を打ち返すことがほとんどできないため，球が自分のバーの奥にある状態を多数訪問することになり，それらの状態の内部報酬が減少していく．逆に相手のバーの奥に球がある状態は目新しい状態であるため，高い内部報酬が得られる．これによりある程度は得点することができるようになる．RND の場合はある程度得点することができるようになり，球が相手のバーの奥にある状態が目新しくなくなった時点で内部報酬が発生しなくなるが，SRG の場合はエピソードが前回よりも 1 ステップ長くなった時点で必ず目新しい系列となるため，エピソードが長くなれば内部報酬が発生し続ける．エピソード長は相手が一方的に得点し続け，スコアが 21 に到達する場合はとても短いものになるが，最終的に負けるとしてもエージェントが得点することでその得点に要したステップ分エピソードを長引かせることができる．したがって SRG はエピソードを長引かせるために自分が得点するようになるため，RND よりもスコアとエピソード長が素早く増加していくと考えられる．

**図 5** は本来の外部報酬，エピソード長，内部報酬の推移である．まず PPO であるが，こちらは通常の報酬設定ではしっかりとスコアを獲得できることが文献 [7] で示されている．しかし，報酬が疎になるように変更した結果，ほとんどスコアを獲得できなくなっているため，1 章で述べた要素 (2) を持つゲームに変化したことが分かる．これに対して SRG は外部報酬，エピソード長，内部報酬がいずれも素早く上昇していることが確認できる．これらの結果より，SRG が想定どおりの挙動を示すことが確認できた．

### 5.2 Graviar

**図 6** より，移動平均ではあまり差は見られなかったが，2500 あたりのスコアを大量に獲得できるのは SRG のみで，PPO と RND ではほとんど獲得できないという結果となった．SRG では行動の目新しさも測るため，様々なタイミングにおける行動を探索することができ，結果的に正しい物体のコントロールを発見することにつながったと考えられる．また，内部報酬は SRG の方が若干多く発生する結果となった．序盤の差の広がりによって探索に差が生まれていることが分かるが，この差がスコアの移動平均の差の広がりに現れていると考えられる．

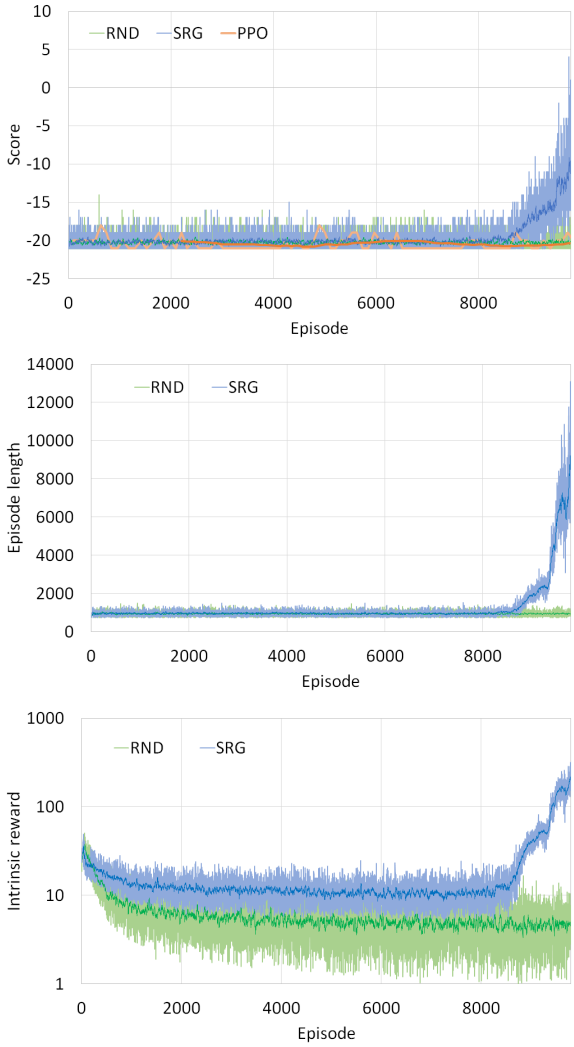



**図 5** Pong における結果．上がスコア（外部報酬），中がエピソード長，下が内部報酬の推移を表す

**Fig. 5** Learning curves of score, episode length and intrinsic reward in Pong.

### 5.3 Montezuma Revenge

**図 7** より，Montezuma Revenge は PPO 以外の両手法でしっかりとスコアを獲得できたことが分かる．また，先に説明した POMDP 性の高い部屋に存在する外部報酬を獲得するまでに要した部屋の訪問回数は，SRG が 126 回，RND が 362 回となり，SRG は RND の約 35%であった．このことから SRG による POMDP 性への対処が効果的であったと考えられる．一方内部報酬を見てみると，SRG は途中で急激に増加している個所が複数あることが分かる．そしてその増加はスコアの増加と同期していることが確認できる．このことから，未知の部屋の訪問に対して敏感に反応していることが分かる．

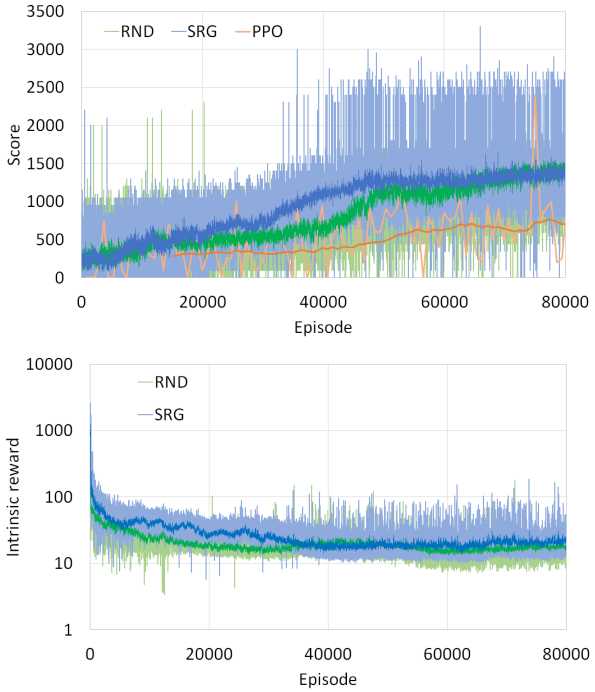



図 6　Gravitar における結果．上がスコア（外部報酬），下が内部報酬の推移を表す

**Fig. 6**　Learning curves of score and intrinsic reward in Gravitar.

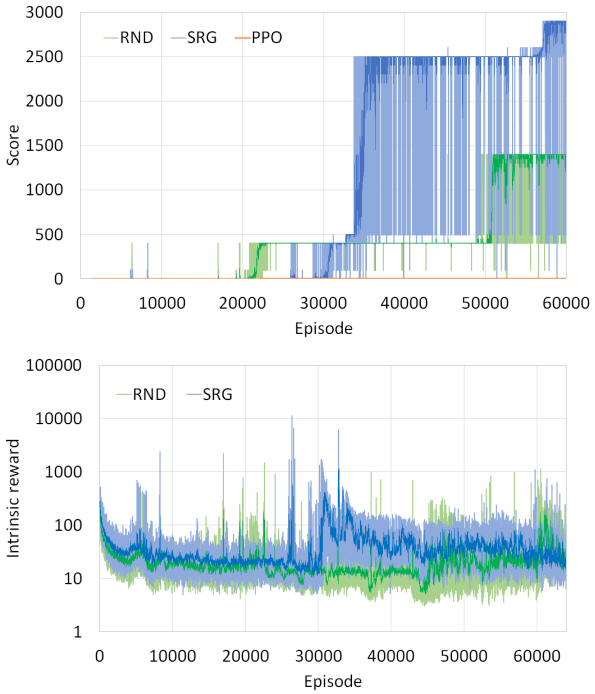



図 7　Montezuma Revenge における結果．上がスコア（外部報酬），下が内部報酬の推移を表す

**Fig. 7**　Learning curves of score and intrinsic reward in Montezuma Revenge.

### 5.4　Pitfall

図 8 を見てみると，すべての手法でまったく正の外部報酬を獲得できなかったことが分かる．SRG によって

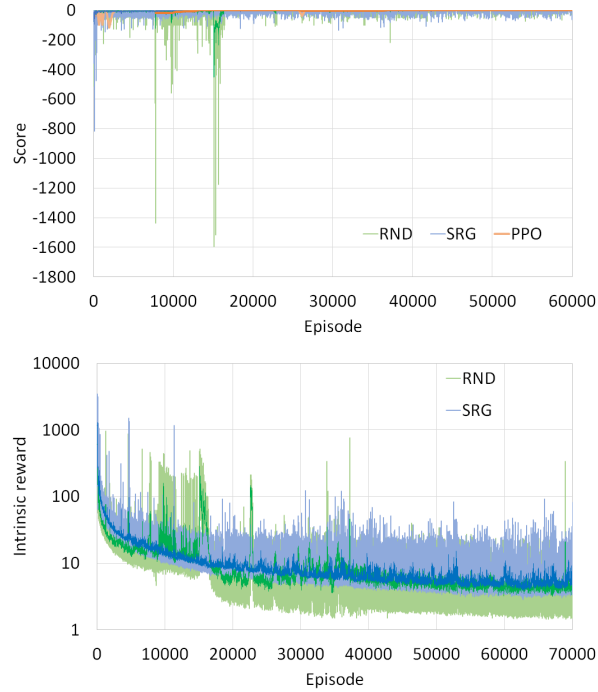



図 8　Pitfall における結果．上がスコア（外部報酬），下が内部報酬の推移を表す

**Fig. 8**　Learning curves of score and intrinsic reward in Pitfall.

POMDP に対応できたとはいえ，それだけでは十分な探索が行えなかったと考えられる．一方内部報酬は 10,000 から 20,000 エピソードの間で RND が大きくなるのと同時に，スコアの方で RND は多くの負の報酬を獲得している．このことから探索しようと様々な部屋を訪問することができているが，その分障害物に接触したと考えられる．SRG は内部報酬が単調に減少しているため，あまり多様な部屋を訪問することができなかったがために障害物に接触する機会も少なく，あまり負の報酬を獲得しなかったと考えられる．

### 5.5　Private Eye

図 9 より，PPO では大きな外部報酬がまったく獲得できなかった．そして好奇心探索を導入した RND と SRG は両手法とも大きな外部報酬を獲得することができたが，SRG は大きな外部報酬を獲得できる頻度が RND よりも高いという結果となった．Private Eye はアイテムを回収する順序が重要なため，元いた部屋に戻らなければならないという状況が存在するが，理論上 RND は訪問済みの部屋へ戻ると内部報酬は発生しないため，そのような動作を行う動機につながりにくい．しかし，SRG では訪問する順序が違えば内部報酬が発生するため，元の部屋へ戻るという動機につながる．このような特性が結果に強く反映されたと考えられる．一方内部報酬ではあまり差異が見られなかったが，序盤で SRG が 4,000 付近のスコアを大量に獲得している段階で SRG のみ内部報酬が上昇していること

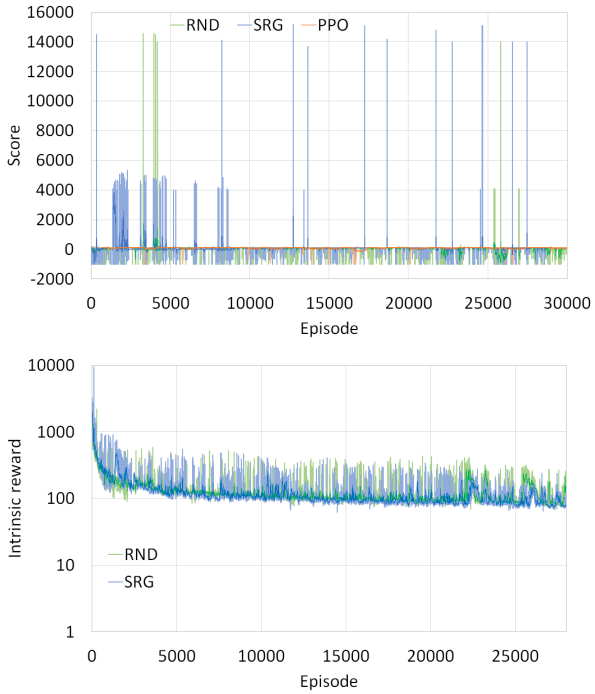



図 9　Private Eye における結果．上がスコア（外部報酬），下が内部報酬の推移を表す

**Fig. 9**　Learning curves of score and intrinsic reward in Private Eye.

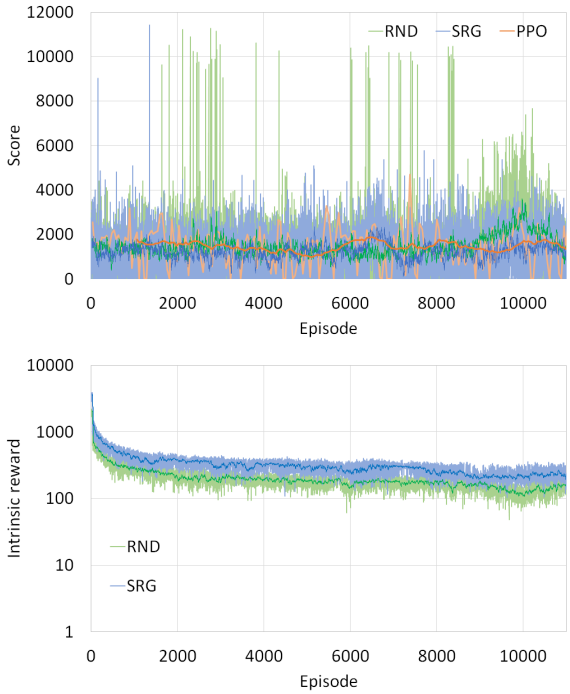



図 10　Solaris における結果．上がスコア（外部報酬），下が内部報酬の推移を表す

**Fig. 10**　Learning curves of score and intrinsic reward in Solaris.

が確認できる．このことから SRG による探索がスコアの獲得につながっていることが分かる．

### 5.6　Solaris

図 10 から，すべての手法でスコアにほとんど差は見られなかったが，内部報酬は SRG の方が RND より若干大きいという結果となった．この環境の POMDP 性が低く，順序関係も特に存在しないという要素から，SRG の特性があまり有効でなかったと考えられる．

### 5.7　Venture

図 11 から，PPO 以外の手法ではしっかりとスコアを獲得することができたが，RND と SRG でほとんど差異が見られないことが分かる．これも Solaris と同じように，SRG の特性が有効でなかったと考えられる．

### 5.8　実験結果まとめ

以上の結果より，好奇心探索を導入することで，通常の深層強化学習ではスコアをあまり獲得できないゲームにおけるパフォーマンスを改善できることが示された．また，RND と SRG を比較した結果，目新しい行動の評価，POMDP 性，順序関係といった SRG の特性が活かせる要素の存在する環境において，学習速度，外部報酬を獲得できる頻度などで改善が見られた．さらに，環境内でも特に POMDP 性が高い場面におけるパフォーマンスの違いなど，局所的な状況における改善も確認できた．一方 Pitfall のように，RND の広範な探索能力が若干損なわれる環境も存在した．

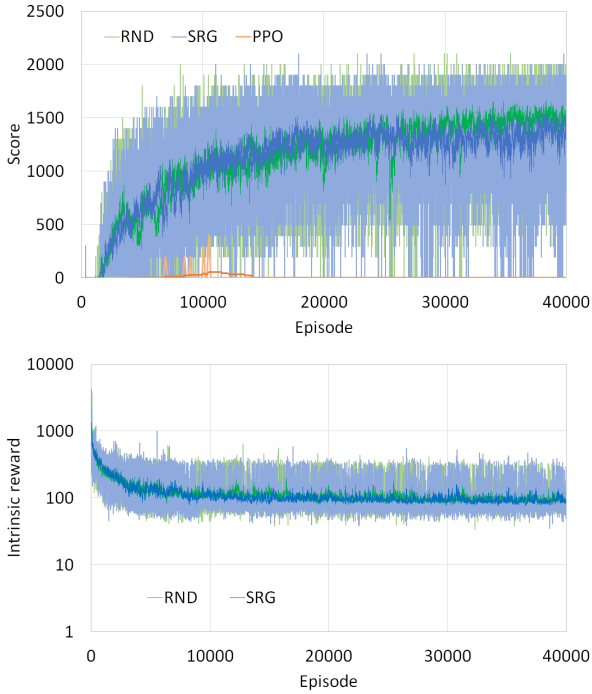



図 11　Venture における結果．上がスコア（外部報酬），下が内部報酬の推移を表す

**Fig. 11**　Learning curves of score and intrinsic reward in Venture.

計算量的な観点から比較すると，SRG は RND にさらに DNN を追加して実現しているため，RND よりも計算量が増大する．しかし，近年の最適化されたライブラリや GPU を用いて実験した結果，このオーバヘッドは特に問題にはならなかった．

## 6. おわりに

本研究では得られた系列の目新しさを測る内部報酬生成器を提案した．提案手法では目新しい状態のみに着目して探索を行う従来手法を，目新しい系列に着目して探索を行うものに拡張し，同時に目新しい行動についての考慮と POMDP の探索へ対応することができるようになった．Atari2600 の報酬設定を変更した Pong と，RND の論文で着目されていた探索の困難な環境における差異を検証した結果，提案手法の特性が活かせる環境や局所的な場面において改善が見られた．

本研究では POMDP に対応した提案手法で Pitfall を学習させたが，結局スコアを獲得することができなかったため，まだ他にも探索に必要な要素が存在すると考えられる．したがって今後の課題としては，Pitfall などの既存手法で解くことのできない環境におけるエージェントの挙動をより詳しく検証し，現状の探索能力に足りない要素を考察することがあげられる．また，本研究では過学習に関する問題は見られなかったが，過学習が容易に起こりうることも事実である．しかし，深層強化学習では事前に DNN に入力されうるデータが存在せず，過学習に関する議論が難しいため，深層強化学習全体の課題といえる．

また，ロボット制御などの現実の環境では，過去の状態に現在の最適行動を決定するうえで重要な情報が含まれていたり，最終的なエージェントの達成目的は簡単に定義できても，その過程にどのような状態を経由すればそれが達成しやすいかが分からない場合が多い．そのため，本研究のように，POMDP への対処，および外部報酬が疎な環境への対策を行うことが，強化学習によって解くことのできる問題をより増やすことにつながると考えられる．

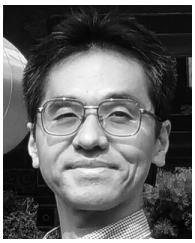
**謝辞**　本研究の一部は，JSPS 科研費 JP19K12118 の助成を受けて行われた．

### 参考文献

[1] Abbeel, P., Coates, A. and Ng, A.Y.: Autonomous Helicopter Aerobatics through Apprenticeship Learning, *International Journal of Robotics Research*, pp.1–31 (2010).
[2] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C. and Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge, arXiv:1409.0575v3 [cs.CV] (2015).
[3] He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition, arXiv:1512.03385v1 [cs.CV] (2015).
[4] 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).
[5] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. and Hassabis, D.: Human-level control through deep reinforcement learning, *Nature*, Vol.518, pp.529–533 (2015).
[6] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J. and Zaremba, W.: OpenAI Gym, arXiv:1606.01540 [cs.LG] (2016).
[7] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O.: Proximal Policy Optimization Algorithms, arXiv:1707.06347 [cs.LG] (2017).
[8] Hausknecht, M. and Stone, P.: Deep Recurrent Q-Learning for Partially Observable MDPs, arXiv:1507.06527 [cs.LG] (2015).
[9] Barto, A.G.: Intrinsic Motivation and Reinforcement Learning, *Intrinsically Motivated Learning in Natural and Artificial Systems*, pp.17–47 (2012).
[10] Burda, Y., Edwards, H., Storkey, A. and Klimov, O.: Exploration by Random Network Distillation, arXiv:1810.12894 [cs.LG] (2018).
[11] Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, *Neural Computation*, Vol.9, No.8, pp.1735–1780 (1997).
[12] 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, arXiv:1602.01783 [cs.LG] (2016).
[13] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *Journal of Machine Learning Research*, Vol.15, No.1, pp.1929–1958 (2014).

**村上 知優**

2018 年名古屋工業大学工学部情報工学科卒業．2020 年同大学大学院情報工学専攻博士前期課程修了．現在，フューチャー株式会社勤務．

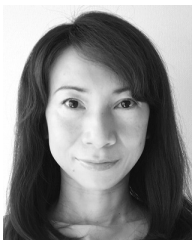


**森山 甲一**

1998 年東京工業大学工学部情報工学科卒業．2003 年同大学大学院情報理工学研究科計算工学専攻博士課程修了．博士（工学）．同専攻助手，大阪大学産業科学研究所助手，助教，特任准教授を経て，現在，名古屋工業大学大学院工学研究科情報工学専攻准教授．人工知能，マルチエージェントシステム等の研究に従事．電子情報通信学会，人工知能学会各会員．

**松井 藤五郎**　（正会員）

2003 年名古屋工業大学大学院工学研究科電気情報工学専攻博士課程修了，博士（工学）．同年東京理科大学理工学部経営工学科助手，助教．2009 年とうごろう機械学習研究所を設立．2010 年中部大学生命健康科学部臨床工学科兼工学部情報工学科講師．2014 年同准教授．強化学習，機械学習，データ・マイニングに関する研究に従事．人工知能学会，AAAI，ACM 各会員．



**武藤 敦子**　（正会員）

1998 年名古屋工業大学工学部知能情報システム学科卒業．同年同大学文部科学技官，2004 年同大学大学院工学研究科助手，2007 年同助教，2016 年同准教授，現在に至る．博士（工学）．人工知能，人工生命，社会ネットワーク分析に関する研究に従事．IEEE シニア会員，人工知能学会，日本数理生物学会各会員．

**犬塚 信博**　（正会員）

1987 年名古屋工業大学工学部卒業．1992 年同大学大学院工学研究科博士後期課程修了．博士（工学）．同年同大学助手．2008 年同大学工学研究科教授．現在に至る．人工知能，特に帰納学習，知識発見，社会ネットワーク分析の研究に従事．人工知能の教育への応用等に興味を持つ．人工知能学会，電子情報通信学会，AAAI，ACM 各会員．