SurpriseとOn-policynessに基づく優先度による省メモリな強化学習

海野良介[1,a)]　鶴岡慶雅[1]

概要：Off-policy 強化学習では，エージェントは環境から遷移データを収集し，パラメータ更新のために集めた遷移データをリプレイバッファに保持する．環境観測が画像で与えられる場合，これらの遷移データを保持するために大量のメモリが消費される．特に，計算資源の限られた状況で強化学習手法を適用する場合，大量のメモリ消費は問題となる．本研究では，遷移データの学習における優先度を計算し，相対的に重要でないと判断されたものから破棄することで，バッファによるメモリ消費を節約する手法を提案する．優先度の評価には，遷移データの *surprise* と *on-policyness* を用いる．Surprise は，データのモデルにとっての目新しさを表し，そのデータから得られる情報を定量化した値であり，On-policyness は，そのデータが現在のモデルの方策に対応しているかを表す値である．本手法により，画像観測の環境において，性能を低下させることなく，リプレイバッファによるメモリ消費を大幅に削減できることを実験的に示した．

Memory-Efficient Reinforcement Learning with Priority based on Surprise and On-policyness

Ryosuke Unno[1,a)]　Yoshimasa Tsuruoka[1]

Abstract: In off-policy reinforcement learning, an agent collects transition data (a.k.a. experience tuples) from the environment and stores them in a replay buffer for the incoming parameter updates. Storing those tuples consumes a large amount of memory when the environment observations are given as images. Large memory consumption is especially problematic when reinforcement learning methods are applied in scenarios where the computational resources are limited. In this paper, we introduce a method to prune relatively unimportant experience tuples by a simple metric that estimates the importance of experiences and saves the overall memory consumption by the buffer. To measure the importance of experiences, we use *surprise* and *on-policyness*. Surprise is quantified by the information gain the model can obtain from the experiences and on-policyness ensures that they are relevant to the current policy. In our experiments, we empirically show that our method can significantly reduce the memory consumption by the replay buffer without decreasing the performance in vision-based environments.

1. はじめに

強化学習 (reinforcement learning, RL) の発展により，視覚的な入力から複雑な動作を学習できるようになった．特に，off-policy 強化学習 [1], [2] は，経験再生 (experience replay) [3] を用いることにより，一般に on-policy な学習よりも高いサンプル効率で学習を行うことが可能である．経験再生では，エージェントが環境で観測した遷移データをタプルとしてリプレイバッファと呼ばれるバッファに保存し，そこからデータをサンプルすることでデータを繰り返し学習に利用する．しかし，環境の観測状態が画像として与えられる場合，画像から適切な表現をモデルが獲得するまでに必要なサンプルが多くなる．そのため，リプレイバッファに保存される遷移データの数量も増え，結果としてモデルの学習に必要なメモリ量が増える．メモリ消費の増加は利用できる計算資源に限りがある場合に問題となる．

[1] 東京大学大学院情報理工学系研究科電子情報学専攻
Department of Information and Communication Engineering, Graduate School of Information Science and Technology, The University of Tokyo
[a)] runno@logos.t.u-tokyo.ac.jp

強化学習におけるリプレイバッファに関する先行研究の多くは，遷移データをバッファからどのようにサンプルするかに注目している [4], [5], [6], [7], [8]. 一方で，利用可能な資源が限られているような状況で強化学習のエージェントを訓練を行おうとする場合，リプレイバッファを適切なサイズに縮小する必要がある．しかし，単純にバッファのサイズを小さくするだけでは，予期せぬ性能低下を招くことが知られている [9], [10]．比較的小さなバッファを用いた強化学習の設定で，新しい遷移データを収集した際にバッファに保存している古い遷移データの中から上書きするものを選択する手法に関する先行研究はある [11], [12], [13] が，メモリ消費量の大きい画像環境において，メモリ効率の良い手法は考慮されていない．そこで，本研究では性能を低下させることなくリプレイバッファのサイズを縮小することを目的とする．

リプレイバッファが消費するメモリ量を削減するために，遷移データの学習における重要度を *surprise* [14] と *on-policyness* [10] に基づいて優先度を計算し，優先度にしたがって遷移データを取捨選択する手法を提案する．Surprise は遷移データのモデルにとっての不確実性に関係し，その情報の目新しさ，その遷移データから得られる情報量を示す．しかし同時に，不確実性の高いデータはモデルにとって外れ値である可能性も高いため，それらの外れ値の悪影響を抑え，現在のエージェントが実際に環境で取る行動に近い遷移データを優先的に保持するために，on-policyness を導入する．本手法は，既存のリプレイバッファの実装を簡単に変更することで実装でき，リプレイバッファから遷移データをサンプリングするために使用される既存のアプローチと組み合わせることが可能である．提案するデータの剪定手法は，リプレイバッファによるメモリ消費を抑え，バッファの容量が制限された場合の性能低下を防ぐことができることを実験的に示す．

2. 関連研究

リプレイバッファを用いた経験再生は，off-policy 強化学習において，行動方策によって収集された遷移データを現在の方策の学習に最大限に活用するための一般的な方法である [1], [2], [15]．リプレイバッファに関する研究の多くは，リプレイバッファから遷移データをどのようにサンプリングしてミニバッチを作成すれば，学習のサンプル効率を向上できるかに焦点を当てている．それらの研究の中で最もよく知られているのは，優先度付き経験再生 (prioritized experience replay, PER) [4] である．PER ではその遷移データを用いて学習を行った際の誤差を用いて，誤差の大きいデータほど頻繁にサンプルされるようにすることで，遷移データを一様にサンプル場合と比較して学習効率の向上を実現した．その他にも，効率的な学習を実現するために遷移データのサンプリングに有効な指標を検討する研究が多くなされている．例えば，固定的な指標を用いて遷移データを選択するもの [4], [6], [16] や，パラメータ更新後のエージェントの性能の改善度を最大にするデータを選択するニューラルネットワークモデルを構築し学習するもの [5], [8] がある．また，比較的小さなバッファを用いてエージェントの学習を行う上で，保持するデータを選択する指標として，報酬 [11]，目新しさ [13]，探索の度合い [13], [17] などの指標によるデータ選択方法がどのように影響するかを調査した研究もある．しかし，これらの研究は比較的低次元の入力環境で検証が行われてきたため，画像といった高次元の入力のための選択肢表として適切かどうかは不明である．また，画像観測の環境において，メモリ効率の良い強化学習アルゴリズムを構築することを目的とした SEER [18] がある．この手法では，学習初期に畳み込みニューラルネットワーク（CNN）エンコーダのパラメータを固定し，観測画像をそのまま保存する代わりに，画像を CNN エンコーダ通した後の潜在表現ベクトルを保存することでメモリ消費を抑えた．この手法と我々の手法との大きな違いは，我々の手法では遷移データの数を明示的に削減するという点である．

提案手法は強化学習の設定においてどの遷移データを保持するかを扱うが，大量のデータの中から保存するデータを選別する手法は，継続的学習 (continual learning) [19], [20], [21], [22] の設定でも見られる．これらの設定では，モデルはデータをストリーム形式で与えられ，オンライン方式で学習が行われる．これらの設定では，学習の初期段階で観測されたデータに関する情報を後続の情報について学習するうちに，モデルが忘れてしまう破滅的忘却 (catastrophic forgetting) [23], [24] が問題となる．この問題を防ぐ方法の一つとして，データの一部をバッファに保存し，各ドメインの要所となる情報を記憶しておくという方法がある．この方法を用いたものでは，データの学習時の勾配 [19], [25]，特徴ベクトル [26]，データの情報量 [27] などの指標を用いて，保持すべきデータを選択したものがある．強化学習にも継続的学習のような設定はあり，これらで学習が進むにつれて異なるドメインのタスクが与えられる [21], [22]．このとき，エージェントは新しいタスクが渡された学習する際に，前のタスクを覚えているかや，前のタスクの知識を用いて素早く適応するできるかといったことが求められる．我々の設定では，単一のタスク領域のみを考慮し，異なるタスクへの適応を必要としないため，これらの手法で求められる破局的忘却への対処はあまり必要でない．

3. 背景

3.1 強化学習

強化学習は機械学習の手法の一つである．他の機械学習の手法である教師あり学習で教師データが入力として与え

られるのとは異なり，強化学習では環境が与えられ，環境との相互作用を通してデータを収集して学習を行う．強化学習の問題設定においてはマルコフ性に従う環境であるマルコフ決定過程 (Markov decision process, MDP) を仮定する．マルコフ性とは遷移先の状態が直前の状態とその時点での行動のみによって依存するような性質である．MDP は以下の構成要素を持つ．

- 状態の有限集合：$\mathcal{S}$
- 行動の有限集合：$\mathcal{A}$
- 状態遷移確率 $p(s'|s,a)$：$\mathcal{S}\times\mathcal{A}\times\mathcal{S}\rightarrow[0,1]$
- 報酬関数 $r(s,a)$：$\mathcal{S}\times\mathcal{A}\rightarrow\mathbb{R}$

時間 t においてエージェントは状態 s_t を環境から受け取り，それに対して方策 $a_t\sim\pi(a_t|s_t)$ にしたがって行動 a_t を選択する．その結果としてエージェントは報酬 r_t と次状態 s_{t+1} を新たに受け取る．この一連の操作を繰り返し，環境から受け取る累積報酬 R を最大化するような方策を求めることが強化学習におけるエージェントの目的である．これは以下の式で表される値を最大化するものである．

$$R=\sum_{t=0}^{\infty}\gamma^t r_t$$

このとき γ $(0\leq\gamma<1)$ は割引率と呼ばれ，将来的に受け取る可能性のある報酬をどの程度重要視するかを決定する定数であり，通常 1 に近い値を用いる．

3.2 Q 学習

Q 学習 [28] は強化学習の代表的な手法である．ある状態 s において行動 a を取り，その後方策 π にしたがって行動を行った際に期待される累積報酬を返す関数を $Q^\pi(s,a)$ とする．この関数を Q 関数と呼び，出力される値を Q 値という．Q 関数は以下のように定式化される．

$$Q^\pi(s,a)=\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t|s_0=s)\right]$$

この Q 関数は以下の TD 誤差 $\delta(\tau)$ と呼ばれる値を最小化するようにして求められる．

$$\delta(\tau)=(Q(s_t,a_t)-Q^{target}(s_t,a_t))^2$$

$$Q^{target}(s_t,a_t)=r(s_t,a_t)+\gamma\max_a Q(s_{t+1},a)$$

Deep Q-Network (DQN) [1] では Q 関数をニューラルネットワークで近似し，学習を行う．分布型 Q 学習 [29] では，TD 誤差は離散確率分布として表される Q 値の分布 $p(Q)$ と目標の Q 値の確率分布 $p(Q^{target})$ との KL ダイバージェンスとして表される．

$$\delta(\tau)=D_{\mathrm{KL}}(p(Q^{target}(s_t,a_t))||p(Q(s_t,a_t)))$$

3.3 経験再生

オンライン学習の設定では，収集されたデータはパラメータの更新に使用されると破棄される．しかし，1 回の更新ループでそのデータを学習することは困難である．また，学習において重要ではあるが，まれにしか現れないような遷移データをすぐ捨ててしまう可能性があり，これにより学習速度の低下を招く可能性がある．経験再生 [3] は，遷移データを格納するためのリプレイバッファを用いる手法である．リプレイバッファは一般的に行動方策によって収集された直近一定数の遷移データを保存するリングバッファとして実装される．リプレイバッファが保存できるデータ数の上限に達すると，最も古い遷移データが破棄され，新たに収集されたデータが追加される．経験値再生は，遷移データを繰り返し利用することで，学習のサンプル効率を向上させる．また，一定数の遷移データを保持することで，オンラインサンプルの時間的相関を抑制する役割や，壊滅的忘却を抑制する役割も担っている．

4. 提案手法

この章では，遷移データを学習における優先度を計算し，算出された優先順位に従って保持するデータを選択する提案手法について紹介する．これは，限られた容量のリプレイバッファに遷移データの数が収まるようにするためである．提案手法では，優先度を計算するための関数 $f(\tau)$ を以下のように定めた．

$$f(\tau_t)=w(s_t,a_t|Q)\delta(\tau_t), \tag{1}$$

ここで，$\delta(\tau)$ は surprise に基づく値で，$w(s_t,a_t|Q)$ は Q 関数を用いて計算した現在のエージェントの方策に基づく on-policyness に関する重みである．それぞれの詳細と提案手法のアルゴリズム全体について，次節以降で説明する．

4.1 Surprise

Surprise とは，その遷移データがモデルにとってどれだけ目新しく，予期せぬものであるかを示す尺度である．強化学習において，モデルにとって未知の状態を効率的に発見するために，surprise に基づくスコアが今まで探索に用いられてきた [4], [30], [31]．また，モデルの学習がまだ十分でない遷移データを優先的にサンプルような，重み付き分布でデータをサンプリングすることで，強化学習における学習速度を高速化することが知られている [2], [4]．このデータの不確実性の尺度として，一般的に TD 誤差が用いられる．誤差が大きいデータは，モデルにとって予期せぬことが多く，その分モデルにとっての学習できる余地が大きい．また，分布型 Q 学習において，TD 誤差は，遷移データ τ から得られる情報利得の近似とみなすことができる．

Algorithm 1 提案手法

Initialize action-value network Q with parameters θ
Initialize replay buffer $\mathcal{D}$ with capacity C
for each time step t **do**
  Select and execute action $a_t \leftarrow \arg\max_a Q_\theta(s_t, a)$
  Receive reward r_t and next state s_{t+1}
  Calculate the pruning priority $f(\tau_t)$ of the tuple τ_t
  if $t \geq C$ **then**
    $\tau^{min} \leftarrow \arg\min_{f(\tau)} \mathcal{D}$
    if $f(\tau^{min}) < f(\tau_t)$ **then**
      $\mathcal{D} \leftarrow \mathcal{D} \backslash \{\tau^{min}\}$
      Store tuple $(\tau_t, f(\tau_t))$ into the replay buffer $\mathcal{D}$
    end if
  else
    Store tuple $(\tau_t, f(\tau_t))$ into the replay buffer $\mathcal{D}$
  end if
  Sample a batch $\mathcal{B}$ from $\mathcal{D}$
  Update θ with $\mathcal{B}$
  Update the priority value in $\mathcal{B}$ and return it to $\mathcal{D}$
end for

$$
\begin{aligned}
I(Q, \tau) &= \mathbb{E}_\tau \left[\mathrm{KL}(p(Q|\tau) || p(Q)) \right] \\
&\simeq \mathbb{E}_\tau \left[\mathrm{KL}(p(Q^{target}) || p(Q)) \right] \\
&= \mathbb{E}_\tau [\delta(\tau)].
\end{aligned}
$$

この 2 行目では，遷移データを与えた元の Q 値の分布が目標 Q 値の分布に近づくという仮定を用いた．これは，Q 値のモデルは遷移データ τ を用いて目標 Q 値との差を最小化するように学習されるからである．すなわち，TD 誤差の大きい遷移データを選択することで，情報利得の大きい遷移データを選択することができる．

4.2 On-policyness

TD 誤差の大きい遷移データのみを残して学習を行うと，学習が不安定になる．これは，現在の方策の状態行動分布から離れた遷移データも学習データに頻繁に現れることがないため，誤差が大きくなる傾向があるためである．この問題を解決するために，優先度の計算にに on-policyness [10] に関する項を追加した．ここで，on-policyness とは，バッファに保存された遷移データがどれだけ現在の方策を反映しているかというものである．

On-policyness は，遷移データを収集した行動方策と，実際にデータを用いて学習を行う現在の方策が同一であること必要とする on-policy 強化学習において重要である．この条件を満たすために用いられる一般的な方法として，学習を行う現在の方策の行動分布がパラメータ更新後も行動方策の行動分布の近くにとどまるように制約するものがある [32], [33]．また，この on-policyness は off-policy 学習においてもモデルの性能を向上させる上で重要であることが知られている [34], [35], [36]．そこで，この on-policyness を優先度の計算に反映させるため，TD 誤差に加え，以下の重み $w(s, a|Q)$ を導入した．

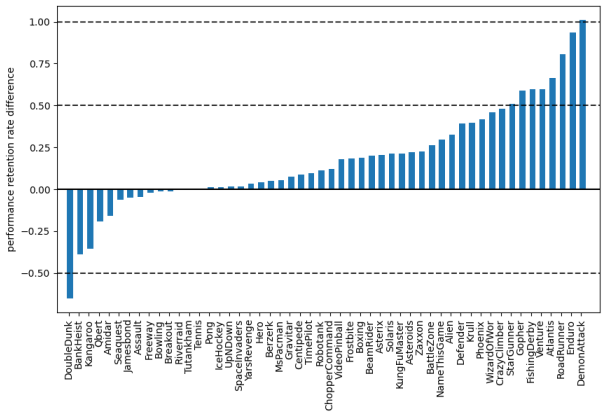


図 1: バッファのサイズに制約した設定における Atari の各環境でのベースラインと提案手法の最高性能の差．制約なしのベースラインの性能で値を正規化している．

$$
w(s_t, a_t|Q) = \frac{\exp(Q(s_t, a_t))}{\sum_a \exp(Q(s_t, a))}
$$

この重み関数 $w(s, a|Q)$ は，ある状態 s_t において行動 a_t をエージェントがどの程度取りやすいか，他の行動と比較してどの程度良いかを反映し，ソフトな方策関数 [37] として見ることもできる．実際の優先度の計算では，新規のデータに対して更に重み付けをするためにに，優先度の初回計算時には重みを常に 1 に設定した．

4.3 アルゴリズム全体

提案手法のアルゴリズムの全体像を Algorithm 1 に示す．遷移データ τ_t が環境から収集された後，遷移データの優先度の値 $f(\tau_t)$ が式 (1) にしたがって算出され，リプレイバッファにタプル $(\tau_t, f^{priority}(\tau_t))$ として追加される．この優先度の値は Q 値の学習の際に TD 誤差が計算される度に更新される．遷移データの剪定は，リプレイバッファに保存されているデータ数が上限に達し，新しい遷移データが収集されたときに実行される．このときに優先度の値が最小の遷移データが破棄される．また，これらの優先度の値は PER で使用されるものとは独立して計算および使用がなされる．

5. 実験

5.1 実験概要

実験は Atari [38] のゲームで行い，提案手法の有効性を検証した．Atari は離散行動の画像観測環境として使われるベンチマーク環境のセットである．本実験では，Atari に含まれる 57 個の環境のうち 52 個を使用した．このうち 5 つのゲームは，外部報酬が疎で大量の探索が必要なため，今回ベースとして用いたアルゴリズムの場合だと制約のない条件でも困難であることに加え，ベースのアルゴリズムが環境から報酬信号を受け取るまでは提案手法自体は探索を促進しないため，これらの環境を評価の対象から除外し

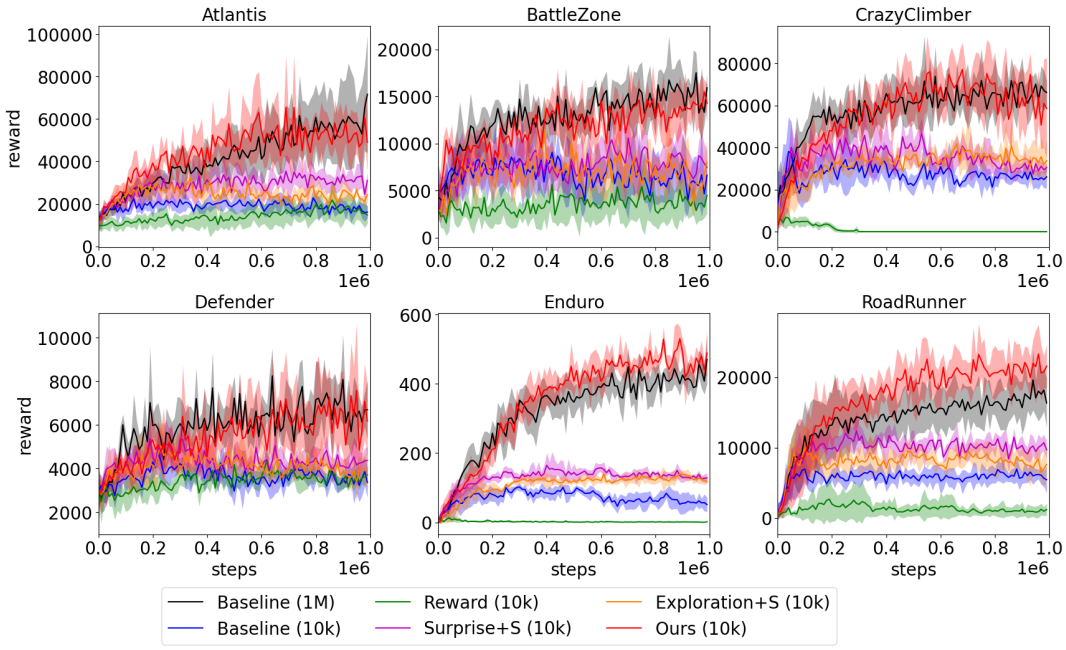


図 2: Atari における各手法の累積獲得報酬の比較．実線と色付き領域は、それぞれ 5 つのランダムシードの結果の平均と標準偏差を表す．

た．エージェントの学習アルゴリズムには Rainbow [2] を使用し，1M ステップ分の訓練をそれぞれ行った．リプレバッファの保持する遷移データの最大数の制約は 10k とした．提案手法を以下のデータ選択手法と比較した．

Baseline：First In First Out（FIFO）方式で遷移データを破棄する．

Reward [11]: 遷移データ内の遷移間で最小の獲得報酬を持つものから破棄する．

Surprise+S [13]: 破棄する遷移データを PER と同様のサンプリング手法で確率的に選択する．サンプリングの優先度は TD 誤差 $\delta(\tau)$ の逆数とした．このサンプリングにより，TD 誤差が小さい遷移データは破棄される可能性がより高くなる．

Exploration+S [13]: 方策の行動確率が高い遷移データから優先的に破棄する．de Bruin らの手法 [13] では，遷移データのある状態におけるそのとき取った行動の確率を直接利用していたが，Rainbow を用いた学習では行動の確率を直接求めることはできない．そのため，提案手法で用いた on-policyness に基づく値をサンプリングの優先度を用いて，破棄するデータを PER と同様の方法でで確率的に選択した．これより，稀な行動選択を行った遷移データはより長くバッファに留まることになる．

また，バッファサイズに制約を設けないベースライン (baseline) 設定についても合わせて実験を行い，性能上限の指標とした．その他のモデルの構成とハイパーパラメータの詳細は付録 A.1 に示す．各環境の実験におけるエージェントの性能評価は 10k 学習ステップごとに 5 回のモデルのロールアウトを実行し，その平均累積報酬を取ることによって行った．各実験は 5 シード分実施した．

5.2 実験結果

Atari 環境でのベースラインと提案手法の比較結果を図 1 に示す．提案手法を用いて遷移データの剪定を行うことで，52 個の環境中 39 個の環境においてリプレイバッファのサイズが制限された場合でも，エージェントの性能の劣化を抑制できた．また，図 2 は，他の遷移データ選択の手法との比較の結果である．提案手法は他の手法の性能を上回っており，制約のないベースラインと同等の結果を得ることができた．また，RoadRunner のような一部の環境では，提案手法を用いることで制約のないベースラインと比較しても高い学習効率を示した．

5.3 各要素の寄与

Surprise と on-policyness のそれぞれが最終的な性能にどの程度寄与したかを調べるために，6 つの Atari 環境において，優先度の計算として片方の要素のみを用いた実験も行った．図 3 の結果から，両方の要素が最終的な性能に寄与していることがわかる．全体的な傾向として，を優先度の計算から on-policyness に関わる項を除外した場合，性能低下がより大きくなった．6 つの環境のうち，Atlantis, BattleZone, Defender, RoadRunner の 4 つでは，on-policyness の指標が最終的な性能達成する上で，十分であるということもわかる．Enduro においては，surprise の項のみでは十分な性能は達成できていないが，取り除くことで学習速度の低下が見られた．

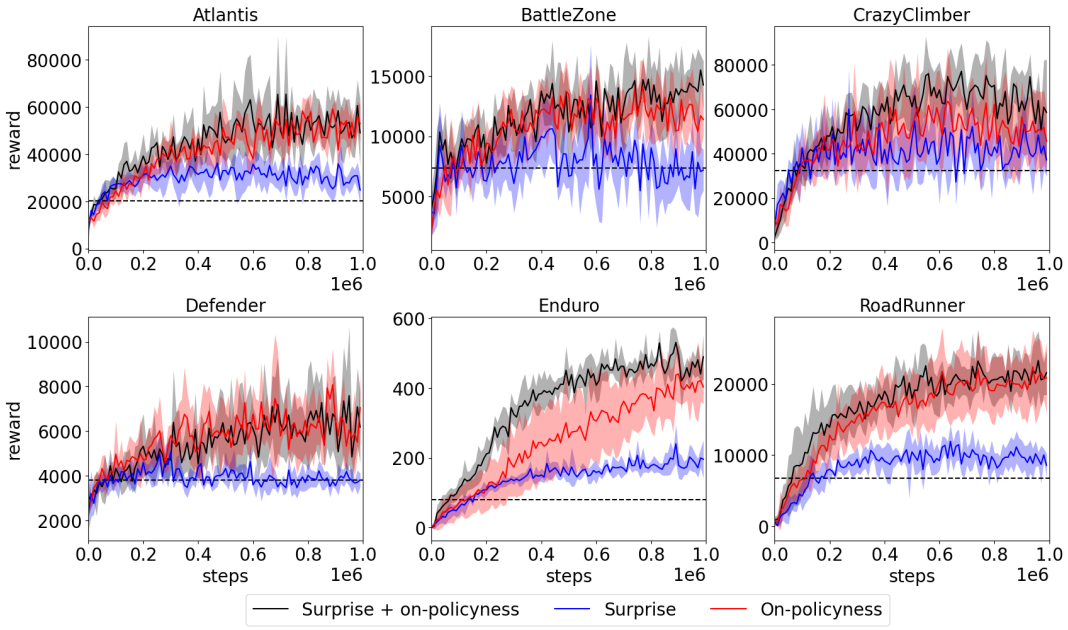


図 3: 優先度の計算から各要素を除外した場合の累積獲得報酬の比較。実線と色付き領域は、それぞれ 5 つのランダムシードの結果の平均と標準偏差を表している．点線は、制約条件下におけるベースライン手法の最高平均スコアを示している．

5.4 バッファ内の状態分布

提案手法においてリプレイバッファに保存された遷移データの状態分布を検証するために，観測画像を CNN エンコーダを通して得られる潜在表現の分布を t-SNE [39] を用いてプロットした．また，遷移内で獲得した報酬も合わせて可視化した．可視化には，バッファの容量に制約のないベースライン，制約のある場合のベースライン及び提案手法のリプレイバッファを用いた．制約がある場合のリプレイバッファは 100k 学習ステップごとに保存し．制約がない設定の場合は，$(100\text{k}\times i+90\text{k}) \sim (100\text{k}\times(i+1))$ $(i=0,\cdots,9)$ ステップ の間に観測された画像状態を使用した．画像の観測状態から潜在表現への変換には，各環境において制約なしベースラインで 1M ステップ学習したモデルの CNN エンコーダを用いた．そのため，同じ環境の異なる実験設定からの潜在表現ではあるが，プロットされた結果は同じ潜在空間を共有している．

状態分布を可視化した結果を図 4 に示す．提案手法が保持する状態遷移はベースラインと比較して，遷移間の累積獲得報酬が大きい傾向があることがわかる．図 4a から提案手法はベースライン手法に比べ，保持している遷移が状態空間に広く分布していることもわかる．また，図 4b の 1M ステップ時の状態分布を比較すると，制約ありのベースライン手法のバッファ内の状態では相対的にあまり分布していない領域に，提案手法が保存する遷移の状態が多く分布していることがわかる．

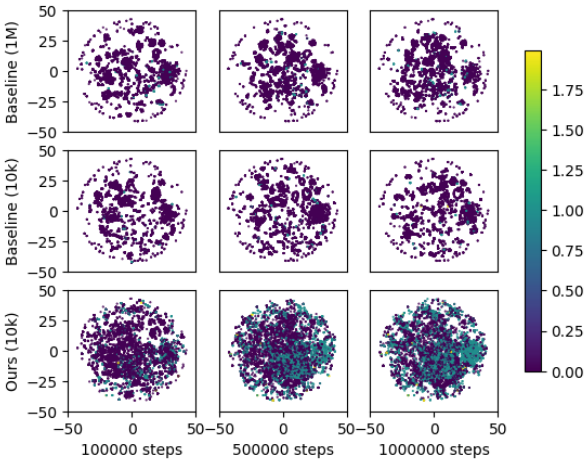
6. 考察

提案手法では，リプレイバッファ内に新たなデータを追加する際に破棄するデータを選択する手法を提案した．本手法によりリプライバッファの容量が制限された場合でもモデルの性能の劣化を抑制することに成功した．Surprise はリプレイバッファが保存する遷移データの状態分布が状態空間内に広く分布することを促進し，on-policyness により遷移データとエージェントの現在のエージェントの方策との結びつきを強いことを保証する．さらに，遷移データが状態空間に広く分布することで，小さなバッファで見られるサンプル間の高い時間的相関を回避する効果があるとも考えられる．

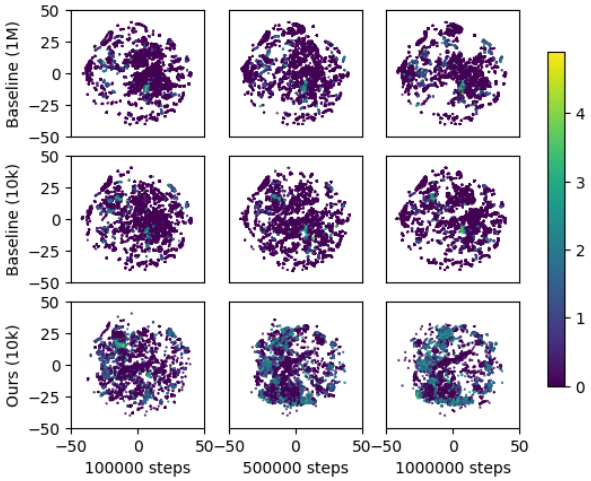
提案手法の限界の一つとして，長期的な学習が不安定であることがある．TD 誤差を遷移データ学習性の指標として用いてきたが，エージェントの学習が進むほど負の影響を与える可能性が大きくなる．これを抑制するために，提案手法では on-policyness の指標を導入することで，外れ値的な遷移データの影響を抑制した．しかし，全体的にベースラインの設定よりも TD 誤差の値が大きくなりやすい傾向があった．また，学習ステップを傘萎えることでエージェントの方策に対応する遷移データほど TD 誤差が小さくなるため，相対的に外れ値的な遷移データの重みが大きくなってしまうため，悪影響を抑制できない場合がある．このような場合の対処法として，学習ステップ数に応じて surprise の影響を小さくすることが考えられる．

7. おわりに

本稿では，遷移データの学習における優先度を学習時の誤差と現在のエージェントの方策との判断の近さに基づいて計算し，それらを用いて保持するべきデータの選択を行



(a) BattleZone



(b) RoadRunner

図 4: t-SNE [39] によって可視化した各学習ステップにおけるリプレイバッファに保存された遷移データの潜在状態分布．それぞれの点の色はその遷移内で獲得した報酬を表している．

うことでリプレイバッファのメモリ消費量を節約する技術を提案した．提案手法を用いることで，リプレイバッファに保存できる遷移データが制限された際に発生するモデル性能の低下を軽減できることがわかった．今度の課題として，今回は扱わなかった連続的な行動空間をもつ環境への本手法の適用や，長いステップ数の学習を行う場合の学習安定化といった点が考えられる．

参考文献

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, Vol. 518, pp. 529–533, 2015.

[2] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[3] Longxin Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, Vol. 8, pp. 293–321, 1992.

[4] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *International Conference on Learning Representations*, 2016.

[5] Daochen Zha, Kwei-Herng Lai, Kaixiong Zhou, and Xia Hu. Experience replay optimization. *International Joint Conference on Artificial Intelligence*, 2019.

[6] Scott Fujimoto, David Meger, and Doina Precup. An equivalence between loss functions and non-uniform sampling in experience replay. *Advances in neural information processing systems*, Vol. 33, pp. 14219–14230, 2020.

[7] Peiquan Sun, Wengang Zhou, and Houqiang Li. Attentive experience replay. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, No. 04, pp. 5900–5907, Apr. 2020.

[8] Youngmin Oh, Kimin Lee, Jinwoo Shin, Eunho Yang, and Sung Ju Hwang. Learning to sample with local and global contexts in experience replay buffer. In *International Conference on Learning Representations*, 2021.

[9] Ruishan Liu and James Zou. The effects of memory replay in reinforcement learning. In *2018 56th annual allerton conference on communication, control, and computing (Allerton)*, pp. 478–485. IEEE, 2018.

[10] William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. Revisiting fundamentals of experience replay. In *International Conference on Machine Learning*, pp. 3061–3071. PMLR, 2020.

[11] Mathijs Pieters and Marco A Wiering. Q-learning with experience replay in a dynamic environment. *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, 2016.

[12] Tim de Bruin, Jens Kober, Karl Tuyls, and Robert Babuška. Improved deep reinforcement learning for robotics through distribution-based experience retention. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3947–3952, 2016.

[13] Tim de Bruin, Jens Kober, Karl Tuyls, and Robert Babuska. Experience selection in deep reinforcement learning for control. *Journal of Machine Learning Research*, Vol. 19, , 2018.

[14] Laurent Itti and Pierre Baldi. Bayesian surprise attracts human attention. In *Advances in Neural Information Processing Systems*, Vol. 18. MIT Press, 2005.

[15] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

[16] Samarth Sinha, Jiaming Song, Animesh Garg, and Stefano Ermon. Experience replay with likelihood-free importance weights. In *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, Vol. 168, pp. 110–123. PMLR, 2022.

[17] Tim de Bruin, Jens Kober, Karl Tuyls, and Robert Babuška. Off-policy experience retention for deep actor-critic learning. In *Deep reinforcement learning workshop, NIPS*, 2016.

[18] Lili Chen, Kimin Lee, Aravind Srinivas, and Pieter Abbeel. Improving computational efficiency in visual reinforcement learning via stored embeddings. *Advances in Neural Information Processing Systems*, Vol. 34, pp. 26779–26791, 2021.

[19] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, Vol. 30, , 2017.

[20] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, Vol. 30, , 2017.

[21] David Isele and Akansel Cosgun. Selective experience replay for lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[22] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, Vol. 32, , 2019.

[23] Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, Vol. 3, No. 4, pp. 128–135, 1999.

[24] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

[25] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, Vol. 32, , 2019.

[26] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

[27] Shengyang Sun, Daniele Calandriello, Huiyi Hu, Ang Li, and Michalis Titsias. Information-theoretic online memory selection for continual learning. *International Conference on Learning Representations*, 2022.

[28] Christopher Watkins and Peter Dayan. Q-learning. *Machine Learning*, Vol. 8, pp. 279–292, 1992.

[29] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pp. 449–458. PMLR, 2017.

[30] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

[31] Glen Berseth, Daniel Geng, Coline Manon Devin, Nicholas Rhinehart, Chelsea Finn, Dinesh Jayaraman, and Sergey Levine. SMiRL: Surprise minimizing reinforcement learning in unstable environments. In *International Conference on Learning Representations*, 2021.

[32] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

[33] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[34] Shangtong Zhang and Richard S Sutton. A deeper look at experience replay. *arXiv preprint arXiv:1712.01275*, 2017.

[35] Matthew Hausknecht and Peter Stone. On-policy vs. off-policy updates for deep reinforcement learning. In *Deep Reinforcement Learning: Frontiers and Challenge*. IJCAI, 2016.

[36] Guido Novati and Petros Koumoutsakos. Remember and forget for experience replay. In *International Conference on Machine Learning*, pp. 4851–4860. PMLR, 2019.

[37] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.

[38] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, Vol. 47, pp. 253–279, 2013.

[39] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, Vol. 9, No. 11, 2008.

[40] Yasuhiro Fujita, Prabhat Nagarajan, Toshiki Kataoka, and Takahiro Ishikawa. Chainerrl: A deep reinforcement learning library. *Journal of Machine Learning Research*, Vol. 22, No. 77, pp. 1–14, 2021.

[41] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

付　　録

A.1　ハイパーパラメータ

学習に用いたモデルの構成は PFRL [40] による Rainbow の実装を用いている．各エピソードは最大 108k フレーム分である．実験におけるハイパーパラメータの一覧を表 A·1 に示す．

表 A·1: Atari の実験におけるハイパーパラメータ

パラメータ名	値
バッチサイズ	32
更新回数毎のステップ数	1
Target 更新ステップ	2000
初期遷移数	1000
Noisy net sigma	0.5
入力画像サイズ	(4, 84, 84)
Reward clipping	True, [-1.0, 1.0]
割引率	0.99
Action repetition	4
学習率	1e-4
Optimizer	Adam [41]
Max gradient norm	10
Multi-step 長	5