# 選択的注意機構を用いたロバストな強化学習手法の実現

岩瀬 諒[1,a)]　鶴岡 慶雅[2]

**概要**：これまでに強化学習は Atari 2600 や囲碁などのボードゲームの分野で人間を超える性能を発揮している．その一方で，同一のタスクであっても学習時からの環境の変動があるような状況にまで汎化して良いパフォーマンスを行うことは困難であり，これは強化学習の課題の一つとして挙げられる．本論文では，画像の一部のみから特徴量を抽出する Attention Agent と呼ばれる手法に，Bisimulation Metrics を用いる強化学習手法である DBC (Deep Bisimulation for Control) によって学習されたエンコーダを組み合わせることによる環境の変動に頑強な強化学習手法を提案する．本稿では提案手法をいくつかのモデル構造で実装し，通常の CarRacing-v0 環境で学習を行なった後に，変化を加えた同環境で汎化性能の検証を行なった．その結果，通常の環境における注目部位の変化を確認することができたが，背景に大幅に変化する環境での汎化性能の向上は見られなかった．そのため，提案手法の問題点について考察し，改善策を議論した．

# A Robust Reinforcement Learning Method Using a Selective Attention Mechanism

Ryo Iwase[1,a)]　Yoshimasa Tsuruoka[2]

***Abstract:*** Reinforcement learning has outperformed humans in games such as Atari 2600 and Go. However, it is difficult to generalize to situations in which the environment has changed since the time of training even for the same task. This is one of the challenges of reinforcement learning. In this study, we propose a reinforcement learning method that is robust to distractions by combining a method called Attention Agent, which extracts features from only a part of an image, with an encoder trained by Deep Bisimulation for Control (DBC), a reinforcement learning method that uses Bisimulation Metrics. In this paper, we implement the proposed method with several model structures, train it in the normal CarRacing-v0 environment, and then verify the generalization capability in the same environment with some modifications. As a result, we confirm the change in the region of attention in the normal environment, although there is no improvement in the generalization capability in the environment with drastic changes in the background. Therefore, we discuss the problem and possible ways to improve proposed method.

## 1. はじめに

人間は環境にタスクに関係のない変動が加わっても，タスクに関係のある部分のみから正しい行動を決定することができる．例えば，車を運転するタスクでは時間や天候が異なっていても，道路の形状や道路標識，周囲の車や歩行者などタスクに関係のある部分に注目し，他の部分を無視することで正しい行動を決定することができる．

しかし，現在の強化学習は環境の変動に脆弱であることが多く，タスクに関係のない部分に変化が加わっただけであっても正しい行動を取ることができなくなることがある．

この問題を解決するために，環境の変動に耐性を持つような強化学習手法の研究が複数行われている [1], [2], [3], [4]. ゲームの分野においても，例えばステージごとに背景が異なるような状況は容易に出現しうるため，やはり環境の変動に耐性のある強化学習手法が求められる．そのような手法の中には，選択的注意 (Selective Attention) の考えに基

[1] 東京大学工学部電子情報工学科
Department of Information and Communication 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
Department of Information and Communication Engineering, Graduate School of Information Science and Technology, The University of Tokyo
[a)] iwase@logos.t.u-tokyo.ac.jp

づいたものがある．選択的注意とは，視界の中でもタスクに関係のある重要な部分にのみ注目し，他の部分は無視するような脳の機能のことであり，これを機械学習の分野に適用した研究は複数行われている [5], [6]．本研究は，注目部位の判別が容易な選択的注意を用いる強化学習手法を用いて，注目部分の判別により比較的高い説明可能性を持ち，タスクに関係のない変動があるような環境においても頑強な動作をすることができる強化学習手法の開発を目的とする．

我々は，選択的注意機構を用いる既存の強化学習手法にタスクに関係のある表現を得られる手法を組み合わせた手法を提案し，変動した環境での汎化性能を計測することで有効性の検証を試みた．その結果，汎化性能の向上は見られなかったものの，タスクに関係のある表現を組み合わせたことによる注目部位の変化を確認した．

## 2. 背景

### 2.1 強化学習

強化学習は環境とのやりとりを経て最適な行動を学習することを目的とする，機械学習の分類の一つである．一般的に，強化学習はマルコフ決定過程 (Markov Decision Process; MDP) によって表される環境を仮定しており，MDP は $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$ のように表される [7]．ここで，$\mathcal{S}$ は状態空間，$\mathcal{A}$ は行動空間，$T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ は遷移関数，$R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathcal{R}$ は報酬関数，$\gamma \in [0, 1)$ は割引率である．強化学習では，エージェントが現在の状態 $s_t$ から行動 $a_t$ を決定し，環境から次状態 $s_{t+1}$ と報酬 $r_t$ を受け取る．これを繰り返したときの累積報酬 $R$ を最大化することが強化学習の目的であり，そのようなときに行動は最適となる．$R$ は式 (1) のように表される．

$$R = \sum_{t=0}^{\infty} \gamma^t r_t \tag{1}$$

環境が変動するような状況においては，MDP の要素の内，状態空間 $\mathcal{S}$，遷移関数 $T$，報酬関数 $R$ が変化すると考えられる．これにより，元の環境で学習したエージェントが変動を加えた環境では正しい行動を選択できないということが起こりうる．

### 2.2 Attention Agent

視界の中でもタスクに関係のある重要な部分にのみ注目し，他の部分は無視するような脳の機能のことを選択的注意という．これは人間が普段行っているようなタスクの解決において重要な役割を担っていると考えられており，選択的注意の概念を強化学習に応用した研究が行われている [8], [9].

その中の一つに，Tang らの研究 [10] がある．Tang らの提案手法 (Attention Agent) のモデル構造を図 1 に示す．

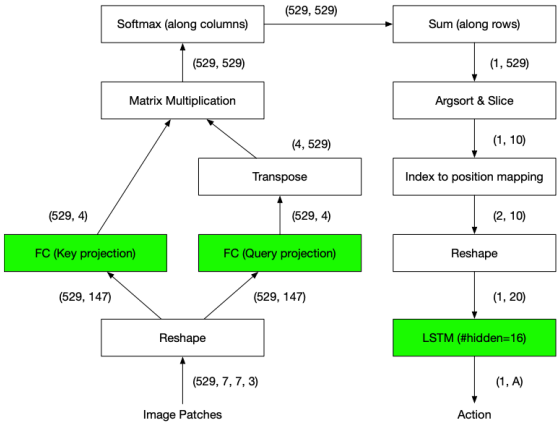



図 1: Attention Agent のモデル構造．更新されるパラメータを持つ層は緑色で示されている．

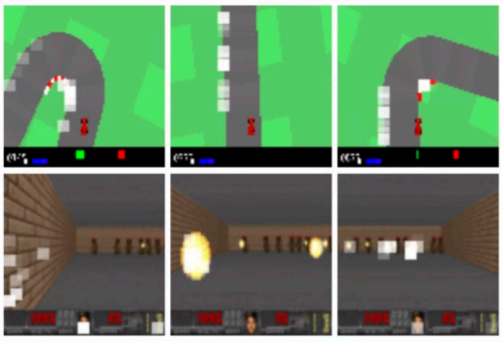


図 2: Attention Agent の注目部位を重ねて表示したタスクの画像．タスクは画像上側が CarRacing-v0，下側が DoomTakeCover.

Attention Agent では，入力画像を切り分けて $N$ 枚のパッチを得たのちに，平滑化したパッチについて self-attention を用いて重要度を計算する．その後，重要度が高い順に上から $K$ 枚のパッチを選んで，それらのパッチから得た特徴量をコントローラへの入力として与えることにより，選択的注意機構を実現している．この際，パッチから得る特徴量としてはパッチの座標を用いている．そのため，コントローラには重要度上位 $K$ パッチの座標が入力として与えられることになる．また，上記の手順は微分不可能であるため，一般的な誤差逆伝播法によるニューラルネットの学習を行うことはできない．そこで，Tang らは共分散行列適応進化戦略 (CMA-ES) [11] と呼ばれる進化戦略に基づいた手法を用いて，self-attention 部分とコントローラの重みの最適化を行なっている．

入力画像をパッチに分割することの利点の一つとして，計算量が比較的小さくなることが挙げられる．画像入力全体に対してピクセル単位で self-attention などの手法を用いて注目部分を求めようとすると，画像のサイズが大きくなるにつれて必要な計算量は著しく増加してしまう．その一方で Tang らの手法では，self-attention への入力は画像

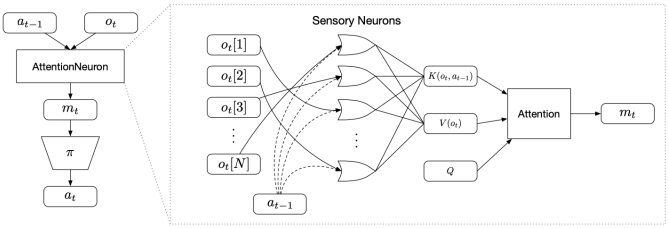



図 3: Attention Neuron のモデル構造.

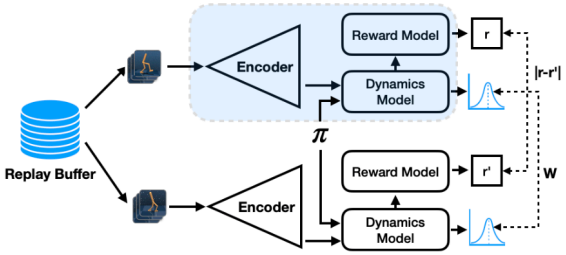



図 4: DBC のモデル構造.

サイズによらず一定であるので，入力画像のサイズが大きくなっても同じ手法で対応が可能であると期待できる．また，強化学習における説明可能性の面でも選択的注意機構を用いることの利点が存在している．つまり，エージェントが注目している部分を知ることにより，エージェントが何を見て判断を行なっているか，人間が通常見るのと同じような位置を見ているかなどを知ることができるようになる．

Tang らによる手法は，CarRacing と DoomTakeCover のタスク及びその一部に変更を加えたタスクにおいて高い性能を発揮している．その一方で，背景をノイズや YouTube のビデオに変化させた CarRacing のタスクにおいては性能の低下が見られ，強い環境の変動に対しては耐性を持たない．これは，入力画像の全体が学習時の入力から著しく変動しているために，各パッチの重要度の計算がうまくいっていないことが原因と考えられる．

### 2.3 Attention Neuron

Tang らによって 2021 年に発表された論文では Attention Neuron というモデルを用いて，状態の入力について Permutation Invariant (PI) な強化学習エージェントを提案している．図 3 に Tang らの提案手法のモデル構造を示す．Attention Neuron は，各観測データと直前の行動をパラメータを共有した Sensory Neuron に入力として与え，その出力について Attention を取って出力としている．図 3 中の Q は入力によらず，状態入力の順番を入れ替えても K と V の行の順番が変わるだけなので，最終的な出力 $m_t$ は状態入力の順番によらず PI となる．この Attention Neuron をポリシーの前段に入れることで PI な特徴量を用いて強化学習を行うことができる．

Tang らの提案手法は PI であるのみならず，環境の変動に対しても耐性があるとしている．論文中では，通常の CarRacing-v0 環境で学習したモデルを再学習無しで背景を未知画像に変更した環境へ適用する実験を行なって汎化性能の検証をしており，高い性能を発揮している．これは，Attention Neuron により計算される特徴量がタスクに関係のない情報を元の画像入力から削減できていることを意味している．しかし，Attention Neuron を用いた手法の検証は背景をノイズやビデオに変えた環境では行われていない．

### 2.4 Deep Bisimulation for Control

Deep Bisimulation for Control (DBC) はタスクに関係のない部分の変動に耐性を持つ強化学習手法の一つであり，Bisimulation Metrics を用いてエンコーダの学習を行い，それにより抽出した表現を入力として下流の強化学習タスクを学習する [1]．Bisimulation Metrics とは，ある 2 つの状態についてそれらから予測される報酬と次状態の確率分布がそれぞれ互いにどの程度類似しているかを示す量であり，直感的には 2 つの状態がどのくらい行動の面で似ている (behaviorally similar) かを表す量である．DBC では学習の際に，この Bisimulation Metrics を用いて行動の面で似ている状態同士を近づけるようにエンコーダの学習を行う．このとき，タスクに関係のない変動があっても Bisimulation Metrics の値は変化しないので，DBC のエンコーダはタスクに関係のない変動に対してロバストな表現を学習することができる．図 4 に DBC の簡単なモデル構造を示す．実際の学習では図 4 の Dynamics Model が次の状態の確率分布を，Reward Model が報酬を予測することで Bisimulation Metrics の計算を行なっている．このときのエンコーダの損失は以下の式で計算される．ただし，$\phi$ はエンコーダを表し，$z_i = \phi(s_i)$ は入力画像をエンコーダに通した出力，$r_i$ は報酬，$\gamma$ は割引率，$\hat{P}$ は状態遷移確率を表す．

$$J(\phi) = \Big( \|z_i - z_j\|_1 - |r_i - r_j| - \gamma W_2 \left( \hat{P}(\cdot|\bar{z}_i, a_i), \hat{P}(\cdot|\bar{z}_j, a_j) \right) \Big)^2 \tag{2}$$

ここで，$W_2$ は 2-Wasserstein 距離であり分布間の距離を計算するために用いられている．式の第二項と第三項はそれぞれ報酬間の距離と遷移する状態の確率分布間の距離を表しており，これらの和は Bisimulation Metrics になっている．このことにより，第一項の状態表現 $z_i$, $z_j$ 間の距離が Bisimulation Metrics に近づくように学習が行われることが分かる．元論文では強化学習手法として Soft Actor-Critic (SAC) を用いて DBC の学習を行なっている [12].

### 2.5 進化的アルゴリズム

進化的アルゴリズムは自然界における生命の進化から着想を得た最適化手法であり，解候補の評価と進化を繰り返して最適化を行う．強化学習の分野においても遺伝的アルゴリズムや進化戦略などの進化的アルゴリズムを用いてネットワークの最適化を行う研究が複数行われており，深層強化学習の手法と比べても劣らない性能を持つとしている研究もある [13], [14]．また，進化的アルゴリズムはロールアウトを並列化することで学習の大幅な高速化を行うことができるという点で優れており，Atari のほとんどのタスクを 1 時間で解くことができるとする研究も存在している [15].

## 3. 提案手法

本研究では，Attention Agent のパッチの重要度の計算にタスクに関係のある表現を用いることで，強い環境の変動に耐性があり，かつ選択的注意機構により注目部位が分かりやすいような強化学習エージェントを提案する．この際，タスクに関係のある表現を抽出するための手法として，Deep Bisimulation for Control (DBC) により学習したエンコーダを用いる．

## 4. 実験

### 4.1 実験概要

本実験では，提案手法について固定したエンコーダを用いる 2 種類のモデル構造とエンコーダの学習を行う 1 種類のモデル構造を用いてそれぞれ強化学習エージェントを構成し，それぞれについて OpenAI gym の CarRacing-v0 環境を用いて学習を行った．その後，背景を変化させた同タスクにおける性能について比較手法として通常の Attention Agent 及び Proximal Policy Optimization (PPO)，通常の CarRacing-v0 環境において DBC を用いて学習した SAC を用いて比較を行った [12], [16].

### 4.2 実験手法

提案手法では，エンコーダから得た特徴ベクトルをどのようにしてパッチの重要度の計算に活用するかが重要である．そこで，提案手法を実装する上で二つのモデル構造を用いて実験を行った．提案手法の具体的なモデル構造を図 6 に示す．図 6(a) のモデルでは，DBC のエンコーダから得た特徴ベクトルを self-attention 部分の key projection 及び query projection の重みに加算して重要度を計算している．一方，図 6 下のモデルでは入力として attention layer のクエリを計算して，パッチから計算したキーと行列積を取ることで重要度を計算している．図 6(b) のモデルは画像から得た特徴ベクトルを用いてパッチ内の注目部位を決定しているという点でより直感的に分かりやすい構造になっている．また，Attention Agent のパラメータ数

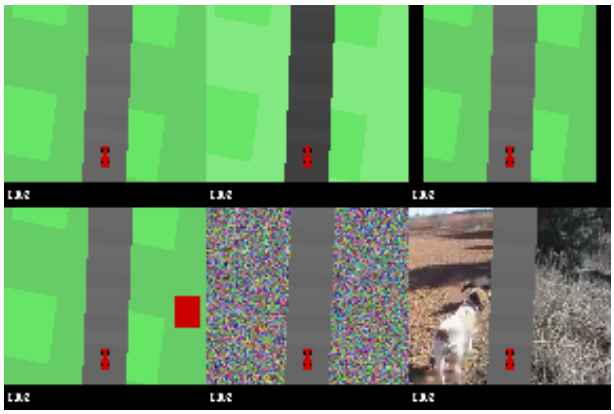
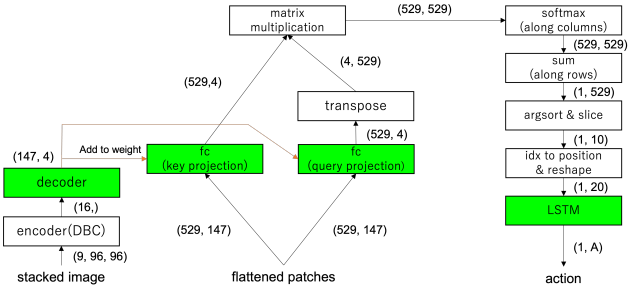
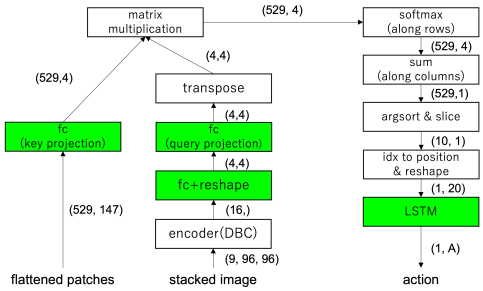


図 5: 変更を加えた CarRacing-v0 環境．左上から順に，通常の環境 (Original)，色を変えた環境 (Color)，画像の端を黒くした環境 (Bar)，赤い塊を描画した環境 (Blob)，背景をノイズにした環境 (Noise)，背景をビデオにした環境 (Video)．ビデオは Kinetics Dataset [17] から walking the dog クラスのものを用いた．




(a) モデル構造 1.




(b) モデル構造 2.

図 6: 提案手法のモデル構造．更新されるパラメータを持つ層を緑色で表している．

3667 に対して，モデル構造 1 はパラメータ数 4268，モデル構造 2 はパラメータ数 3367 となっている．

モデル 1，2 では，事前に学習した DBC のエンコーダを固定して Attention Agent 部分のみを CMA-ES で学習している．これらの手法においては，DBC によって学習されたエンコーダから得られる表現を Attention Agent のパッチの重要度の計算に用いることで汎化性能が向上することが期待される．その一方で，モデルの性能がエンコーダの性能によって制限されるのではないかという懸念が存

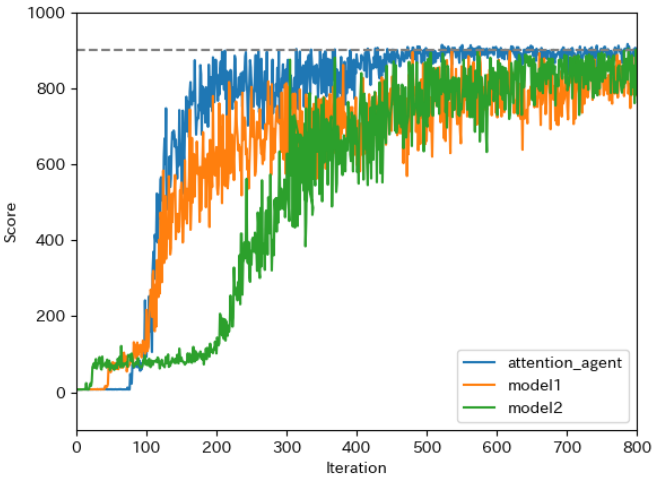



図 7: 通常の CarRacing-v0 環境で Attention Agent と提案手法のモデルを学習した際の学習曲線．提案手法は Attention Agent と比べて学習に時間がかかっている．

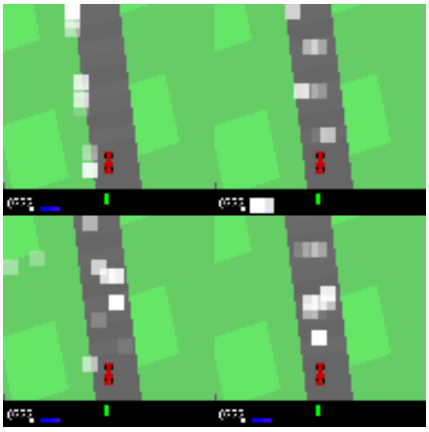


図 8: 通常の CarRacing-v0 環境で Attention Agent (左上), Model1 (右上), Model2 (左下)，DBC+AA (右下) のそれぞれの注目部位を重ねて表示したタスクの画像．

在する．

そこで，提案手法の異なる実装として DBC のエンコーダと Attention Agent 部分を同時に学習するようなモデル構造を作成した．このモデルの構造は図 4 のモデル構造において方策 $\pi$ を Attention Agent に置き換え，エンコーダの入力をパッチに変えたものになっている．このモデルは平滑化したパッチをエンコーダの入力として与え，エンコーダから得られる表現を Attention Agent の入力としている．また，エンコーダの学習のための補助的なネットワークとして Transition Model と Reward Model を用いる．これらによって次状態の確率分布と報酬が予測され，式 (2) の損失を用いてエンコーダの学習を行う．元の DBC が画像全体から表現を学習するのに対して，こちらのモデルではパッチ毎に表現の学習を行う．このモデルを DBC+AA と呼ぶことにする．Attention Agent 部分の学習は CMA-ES，エンコーダと Transition Model，Reward Model の学習は Adam を用いて行う．

これらのモデルを用いて CarRacing-v0 環境において学習を行い，その後背景を変更した CarRacing-v0 環境において 100 エピソードの平均スコアを計算することで，モデルの汎化性能を計測した．背景を変更した CarRacing-v0 環境の画像を図 5 に示す．CMA-ES の集団サイズは 64，初期の分散は 0.1 に設定し，各個体の評価は 4 回のロールアウトの平均値として最適化を行なった．

### 4.3 実験結果

#### 4.3.1 通常の CarRacing-v0 環境における提案手法の学習

図 6 の 2 つのモデル構造を用いて CarRacing-v0 環境で学習を行なった結果を図 7 に示す．CarRacing-v0 環境ではスコアが 900 以上でタスクが解けたとみなすことができる．図 7 から，提案手法の学習は Attention Agent と比べてサンプル効率が低いことが見て取れる．これは，モデルのパラメータ数と入力データの次元数が増加したことによるものであると考えられる．また，最終的なスコアにおいても提案手法は Attention Agent より若干低い結果となった．

次に，提案手法と Attention Agent の通常の CarRacing-v0 環境での注目部位を図 8 に示す．図 8 において，通常の Attention Agent が道の境界に注目している一方で，Model1，Model2，DBC+AA は明らかに道の内部に多くの注目部位を持っている．このことから，モデルに DBC のエンコーダを用いることにより重要度の計算に影響を与えられていると考えられる．

#### 4.3.2 汎化性能の検証

変更を加えた CarRacing-v0 環境における提案手法と各比較手法の性能の検証結果を図 9 に示す．図 9 より，モデル 1 は Original, Color, Bar, Blob の簡単な環境では Attention Agent に近い性能をしているが，本来の目的である Noise と Video のタスクについての改善は見られない．また，モデル 2 については Noise と Video のタスクに加えて，Blob と Color のタスクにおいても性能の低下が見られ，モデル 1 と同様に汎化性能の向上は見られない．

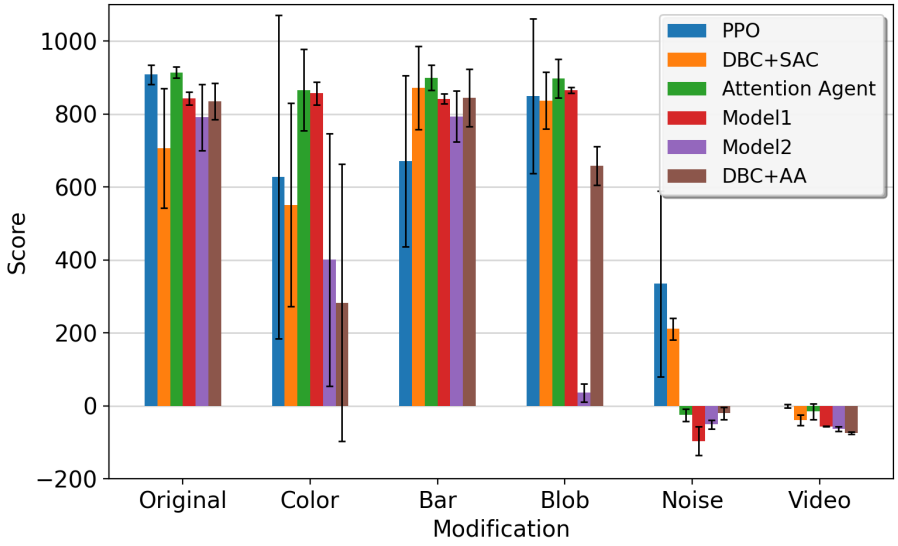



図 9: 変更を加えた CarRacing-v0 環境における各手法のスコア．比較手法としては PPO，DBC を用いて学習した SAC，Attention Agent を用いた．値は各手法，各環境毎に 100 回の試行についての平均値と標準偏差．

DBC+AA についても，Noise と Video のタスクでは性能の向上は見られない．

次に，背景をビデオにした環境における各手法の注目部位を図 10 に示す．図 10 を見ると，Attention Agent と提案手法のどちらもほとんどの注目部位がタスクと関係のない部分になっており，正常に動作していないことが見て取れる．かろうじて Model2 の注目部位は道路上に多く位置しているように見えるが，図 9 を見てもスコアは低いためこれはあまり性能に寄与していないと考えられる．

### 4.4 考察

実験結果から，提案手法の汎化性能は Attention Agent と比べて改善していないことが確認された．この原因と改善策について考察する．

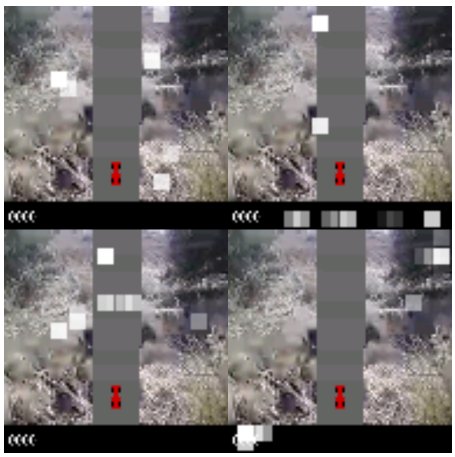


図 10: 背景をビデオにした CarRacing-v0 環境で Attention Agent (左上)，Model1 (右上)，Model2 (左下)，DBC+AA (右下) のそれぞれの注目部位を重ねて表示したタスクの画像．

提案手法の汎化性能が向上しない原因としてまず考えられるのが，DBC によって学習されたエンコーダがタスクに関係のある表現をうまく抽出することができていないという点である．これについては図 9 を見ると分かるように，DBC+SAC の性能が Noise，Video 環境で低下していることから要因の一つであると考えられる．しかし，DBC+SAC が Noise 環境ではある程度の正のスコアを記録しているのに対して，そのエンコーダを利用した Model１，Model2 は Noise 環境では 0 付近のスコアであることから，他にも要因が存在していると考えられる．DBC が有用な表現をうまく抽出することができていない問題の解決策の一つとして，学習時にも変動を加えた環境を用いることが挙げられる．現在の問題設定では，通常の環境で学習を行い再学習なしで様々に変化した環境への汎化を行うことを目的としている．そのため，学習用の変動を加えた環境をこちらから用意することは問題設定に反するために望ましくないが，例えば，モデル内部に学習時の入力に適当な変化を加えて入力とするような機構を加えることで性能が向上する可能性がある．

次に考えられるのは，パラメータ等を含めたモデル構造の問題である．つまり，エンコーダから得られた表現に問題はなくても，モデルがその表現を適切に活用する構造になっていないため学習に悪影響を与えている可能性がある．この点について検証するためには，モデル構造やパラメータを変えて実験の試行回数を重ねることが必要だと思われるが，現在の提案手法の実装では学習にかなりの時間がかかってしまい効率が悪い．そのため，学習時間についても考慮したモデル構造を考案する必要がある．

他の改善策として，Attention Agent のコントローラの入力が現在はパッチの座標のみであるのでパッチそのものを入力に加えること，表現を学習する手法として DBC 以外の手法を用いることなどが考えられる．

## 5. おわりに

本稿では，Attention Agent と DBC を組み合わせた強化学習手法の提案を行った．今回行った実験の中では，提案手法が Attention Agent を上回る汎化性能を持つことは確認できなかった．そのため，大幅に変化するような環境に対する汎化性能を向上させるようなモデル構造を考案することは今後の課題として残っている．また，本実験では一つのタスクのみを用いて学習と汎化性能の検証を行ったが，今後は異なる環境にも提案手法を適用して検証を行うことも考えられる．

### 参考文献


[1] Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *ICLR*, 2021.

[2] Shani Gamrian and Yoav Goldberg. Transfer learning for related reinforcement learning tasks via image-to-image translation. In *ICML*, pages 2063–2072. PMLR, 2019.

[3] Xiang Fu, Ge Yang, Pulkit Agrawal, and Tommi Jaakkola. Learning task informed abstractions. In *ICML*, pages 3480–3491. PMLR, 2021.

[4] Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. Network randomization: A simple technique for generalization in deep reinforcement learning. In *ICLR*, 2020.

[5] Marijn F. Stollenga, Jonathan Masci, Faustino J. Gomez, and Jürgen Schmidhuber. Deep networks with internal selective attention through feedback connections. In *NIPS*, pages 3545–3553, 2014.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[7] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, and Joelle Pineau. An introduction to deep reinforcement learning. *arXiv preprint arXiv:1811.12560*, 2018.

[8] Liu Yuezhang, Ruohan Zhang, and Dana H Ballard. An initial attempt of combining visual selective attention with deep reinforcement learning. *arXiv preprint arXiv:1811.04407*, 2018.

[9] Alex Mott, Daniel Zoran, Mike Chrzanowski, Daan Wierstra, and Danilo J Rezende. Towards interpretable reinforcement learning using attention augmented agents. In *NeurIPS*, 2019.

[10] Yujin Tang, Duong Nguyen, and David Ha. Neuroevolution of self-interpretable agents. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pages 414–424, 2020.

[11] Nikolaus Hansen. The cma evolution strategy: A tutorial. *arXiv:1604.00772*, 2016.

[12] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, pages 1861–1870. PMLR, 2018.

[13] David E Moriarty, Alan C Schultz, and John J Grefenstette. Evolutionary algorithms for reinforcement learning. *Journal of Artificial Intelligence Research*, 11:241–276, 1999.

[14] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv:1712.06567*, 2017.

[15] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

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

[17] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv:1705.06950*, 2017.


## 付　　録

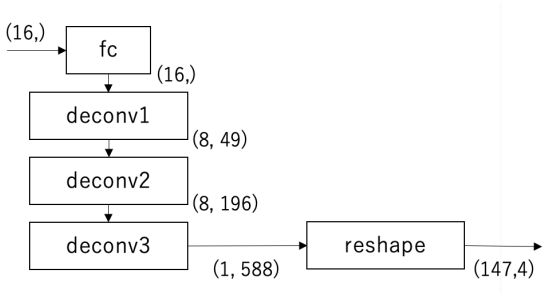



図 A·1: 図 6(b) の decoder の構造.