

2024年度
中部大学大学院工学研究科情報工学専攻

博士学位論文

深層強化学習エージェントの性能向上と
意思決定に対する視覚的説明に関する研究

板谷 英典

論文要旨

深層強化学習とは、エージェントが環境とのインタラクションを通じて試行錯誤を繰り返しながら、エージェントの最適な振る舞いを学習する技術である。この深層強化学習エージェントは、高い制御性能を獲得できることが知られており、ビデオゲーム攻略やロボティクスなど動的な制御が求められるタスクへの応用が期待されている。しかしながら、深層強化学習にはエージェントモデルの性能と信頼性に対する課題が存在する。

エージェントモデルの性能に対する課題として、学習効率の低さが挙げられる。これは、環境とのインタラクションを通じて学習データを収集するため、エージェントモデルの学習に大幅な計算資源と時間を要してしまうことである。この課題に対して、マルチタスク学習の一種である補助学習の導入によるメインタスクの学習効率向上が報告されている。補助学習とは、メインタスクとは異なる目標を持つ補助タスクを、メインタスクと同一モデルで学習することで、メインタスクの学習を促進させるアプローチである。しかし、補助タスクは必ずしもメインタスクに有効であるとは限らず、あらかじめメインタスクに適した補助タスクを人手で選定する必要がある。この人手による補助タスクの選定には、明確な基準が存在せずメインタスクごとに有効な補助タスクが異なるため、どのような補助タスクがメインタスクに有効か判断することが非常に困難である。深層強化学習エージェントの高性能化の背景には、深層学習の恩恵が大きく関わっており、エージェントモデル内部のパラメータ増加と複雑化に伴い、エージェントモデルの意思決定プロセスに対する判断根拠がブラックボックスとなっている。この課題は、エージェントモデルがなぜそのような行動を選択したか、エージェントモデルの判断根拠がユーザには理解できないことである。これは、深層強化学習エージェントに対する信頼性に直結する重大な課題である。

本研究では、はじめにメインタスクに適した補助タスクの選定に関する問題を解決する。この問題の解決に向け、深層強化学習による補助タスクの動的な選択を提案する。補助タスクの損失に対するバイナリ重みを深層強化学習によって制御することで、補助タスクの選択を実現する。本手法は、メインタスクの学習段階に合わせた補助タスクの動的な選択を実現し、メインタスクに対するエージェントモデルの性能向上に貢献する。次に、深層強化学習エージェントモデルのブラックボックス問題を解決する。この問題の解決に向け、エージェントモデルの方策と状態価値、または選択可能な行動全てに着目したエージェントモデルの意思決定プロセスに対する視覚的説明を提案する。方策と状態価値に着目した視覚的説明では、Actor-Critic ベースな深層強化学習手法を対象とし、出力ブランチに attention 機構を導入する。これにより、方策と状態価値の2つの観点からエージェントモデルの意思決定に対する視覚的な解析を可能とした。選択可能な行動全てに着目した視覚的説明では、行動情報を query とする Transformer encoder-decoder 構造をもとにエージェントモデルを構築する。これにより、行動ごとに固有の attention weight を獲得可能とし、エージェントモデルの意思決定に対する網羅的な解析を実現した。一方で、これら視覚的説明では最終的なエージェントモデルの意思決定に対する洞察は、ユーザの主観に依存してしまう。そこで、前述の視覚的説明情報にもとづく大規模言語モデルによる深層強化学習エージェントモデルに対する言語的説明を提案し、視覚的説明から言語的説明への拡張について議論する。

目次

第 1 章 序論	1
1.1 研究の背景	2
1.2 研究目的	3
1.3 本論文の構成	4
第 2 章 深層強化学習の研究動向	7
2.1 強化学習	7
2.1.1 強化学習アルゴリズムの分類	10
2.1.2 Value ベースにおける代表的な深層強化学習手法	14
2.1.3 Policy ベースにおける代表的な深層強化学習手法	18
2.1.4 Actor-Critic ベースにおける代表的な深層強化学習手法	20
2.1.5 Transformer 構造を導入した深層強化学習手法	22
2.2 深層強化学習の応用事例	26
2.3 深層強化学習の課題	27
2.4 説明可能な強化学習に関する研究動向	30
2.4.1 エージェントモデルに対する視覚的説明	31
2.4.2 エージェントモデルに対する言語的説明	35
第 3 章 深層強化学習による補助タスクの動的な選択	37
3.1 深層強化学習による補助タスクの選択モジュール	38
3.1.1 損失関数	38
3.1.2 学習アルゴリズム	39
3.2 評価実験	40
3.2.1 ゲームスコアによる性能比較	41
3.2.2 選択された補助タスクの分析	42
3.2.3 学習段階に応じた補助タスクの動的選択	44
3.3 まとめ	45
第 4 章 方策と状態価値に着目した深層強化学習エージェントに対する視覚的説明	46
4.1 Mask-Attention A3C	47

4.1.1	Mask A3C モデルの構造概要	48
4.1.2	Mask-attention Loss	49
4.2	評価実験	50
4.2.1	ビデオゲーム攻略タスクにおける実験詳細	50
4.2.2	mask-attention に対する Mask-attention Loss の効果	52
4.2.3	従来手法を用いた注視領域可視化による定性的比較	52
4.2.4	mask-attention の可視化によるエージェントモデルの意思決定プロセスに対する分析	54
4.2.5	Atari 2600 におけるスコア比較	58
4.2.6	Inverse mask-attention を用いたスコア減少による比較	60
4.2.7	mask-attention に着目した新しい状態に対するエージェントの反応	61
4.2.8	ロボットマニピュレーションタスクへの応用	63
4.2.9	ユーザによるエージェントモデルの行動予測にもとづく mask-attention の解釈性評価	67
4.3	まとめ	70
第 5 章	選択可能な行動に着目した深層強化学習エージェントに対する視覚的説明	72
5.1	Action Q-Transformer	73
5.1.1	Feature extractor	73
5.1.2	Transformer Encoder-Decoder	73
5.1.3	Action branch	75
5.1.4	Target Trained Q-Network	76
5.2	AR インターフェースを用いたユーザに対する視覚的フィードバック	76
5.3	評価実験	77
5.3.1	ビデオゲーム攻略タスク	77
5.3.2	ロボット制御タスク	83
5.4	まとめ	90
第 6 章	視覚的説明から言語的説明への拡張	92
6.1	視覚的説明情報を活用した LLM による言語的説明	93
6.1.1	Domain knowledge database	94
6.1.2	LLM による説明文生成のためのプロンプトエンジニアリング	94
6.2	評価実験	97
6.2.1	Breakout における言語的説明	97
6.2.2	Seaquest における言語的説明	101
6.2.3	Boxing における言語的説明	103
6.2.4	言語的説明を活用したユーザに向けたマニュアル作成	104
6.3	まとめ	107

第7章 結論と展望	108
7.1 結論	108
7.2 展望	109
謝 辞	111
参考文献	112
研究業績一覧	126

目次

1.1	本論文の構成.	6
2.1	強化学習の概要	8
2.2	強化学習のアルゴリズムマップ (方策の表現方法による分類)	11
2.3	Actor-Critic 法の構成	14
2.4	確率密度関数	15
2.5	DQN のネットワーク構造	16
2.6	Rainbow における 6 つの要素の影響 (文献 [1] から引用)	17
2.7	A3C の構造	21
2.8	Transformer アーキテクチャ (文献 [2] から抜粋)	23
2.9	Decision Transformer の概要 (文献 [3] から引用)	25
2.10	UNREAL のネットワーク構造	29
2.11	Greydanus らの視覚的説明手法における摂動画像例 (文献 [4] から引用, 式 (2.37) 参照)	33
2.12	Mott らの視覚的説明手法におけるエージェントモデル構造 (文献 [5] から引用)	34
3.1	UNREAL + Auxiliary Selection におけるエージェントモデルの概略. ここで Pixel Control, Value function Replay, Reward Prediction は Jaderberg らによって提案された教師なし補助タスクである.	38
3.2	Deepmind Lab における 3 つのゲームとエージェントの観測情報と行動情報. 右下はエージェントの行動を示し, 本実験で用いるゲームタスクでは, jump, crouch, rotate up/down は使用しない.	41
3.3	DeepMind Lab の 3 タスクにおける学習時の報酬推移.	42
3.4	1 エピソード間における Auxiliary Selection の行動ごとに対する選択回数.	43
3.5	補助タスクの組み合わせパターンを追加した horseshoe における学習時のスコア推移.	44
3.6	学習段階に応じた 1 エピソード間における補助タスクの選択率 (seekavoid).	45
4.1	Mask-attention Loss を導入した Mask-attention A3C の概略	47
4.2	Mask-attention Loss による mask-attention の比較. 矢印はパックマンの進行方向を示す.	52

4.3	エージェントモデルの注視領域可視化に関する Greaydanus らの手法との比較. ここで, Ours は Mask A3C MaskattL による mask-attention の可視化例である.	53
4.4	Policy branch における mask-attention の可視化例. 可視化結果は, Mask A3C MaskattL を用いた mask-attention の可視化例である. Image with mask-attention 下部のコントローラは, 現状態で深層強化学習エージェントが選択した行動を示す.	55
4.5	Value branch における mask-attention の可視化例. 画像は, Mask A3C MaskattL を用いた mask-attention の可視化例である. 状態価値が大きく変化する 2 フレームでの mask-attention の例を示す. 右のグラフは状態価値の遷移を示し, グラフ中の緑破線は各ゲームにおける次ステージへの遷移を示す.	56
4.6	inverse policy mask-attention による平均スコアの低下率. ここで, 平均スコアは 100 エピソード間における平均スコアである.	60
4.7	mask-attention による新しい状態へのエージェントの反応の可視化. これは Mask A3C MaskattL を用いた mask-attention の可視化である.	62
4.8	ロボットマニピュレーションにおける Mask A3C の概要.	64
4.9	ロボットマニピュレーションタスクにおける mask-attention の可視化例. Ours は Mask PPO MaskattL による mask-attention の可視化である.	66
4.10	把持対象物体の予測に対する設問の回答詳細	70
5.1	Action Q-Transformer 構造の概要	73
5.2	Transformer Encoder-Decoder の詳細	74
5.3	Query branch の詳細	75
5.4	AR を用いた attention weight の可視化例.	77
5.5	Atari 2600 における 100 エピソード間の平均スコア. グラフは, ベースラインである Rainbow モデル (赤破線) の平均スコアを 100 としたときの, 各手法の平均スコアに対する割合を示している.	79
5.6	Breakout における attention weight の可視化例. 画像上の白矢印は, ボールの進行方向を示している.	80
5.7	Seaquest における attention weight の可視化例	82
5.8	Unity で作成された屋内環境におけるロボット制御シミュレーション.	84
5.9	ロボット制御タスクにおけるゴール情報を考慮した Query branch.	85
5.10	ロボットの自律制御タスクにおける decoder-attention の可視化例. 入力画像は (a),(b) ともに同じである.	87
5.11	ロボット動作に対する予測設問に関する調査のフローチャート. 右に練習設問例を示す.	88
5.12	実世界にロボットエージェントの振る舞いに対する視覚的フィードバック例. 左に定点カメラによる動画例, 右に AR デバイスを用いた attention weight 可視化動画例を示す. また, 白枠で囲われた領域には HSR が存在し, 白矢印は HSR の進行方向を示している.	89

5.13	ロボットの振る舞いに対するユーザの主観的理解度アンケートの回答結果. 各数値は回答者の割合を表す.	90
6.1	深層強化学習エージェントモデルの意思決定に対する言語的説明手法の概略	93
6.2	ステップ・バイ・ステッププロンプトの概要.	95
6.3	短期説明と長期説明プロンプトの例.	96
6.4	Breakout におけるパドルがボールに向かうシーンに対する言語的説明例. ゲーム画面上の白矢印は, ボールの進行方向を示している.	98
6.5	Breakout におけるボールの打ち返しに失敗したシーンに対する言語的説明例. ゲーム画面上の白矢印は, ボールの進行方向を示している.	99
6.6	Breakout におけるパドルにボールが向かうシーンに対する言語的説明例 (w/o attention weight). ゲーム画面上の白矢印は, ボールの進行方向を示している.	100
6.7	Seaquest における言語的説明例. 短期説明プロンプトは, 異なる 2 シーンでの言語的説明例を示す.	102
6.8	Boxing における敵プレイヤーに攻撃するシーンに対する言語的説明例.	104
6.9	言語的説明を活用したユーザに向けたマニュアル作成の概略	105
6.10	Space Invaders における言語的説明を活用したマニュアル作成例.	106

表目次

2.1	説明可能な強化学習に関する研究のカテゴリライズ	31
3.1	1 エピソード間における各補助タスクの選択回数. 補助タスクの選択回数は, 50 エピソードの平均値である.	43
4.1	ビデオゲーム攻略タスクにおける実験条件	51
4.2	Breakout における 1 フレームあたりに対する注視領域の計算時間とエージェントモデルサイズ. ここで A3C の計算時間は, 方策と状態価値を計算する推論時間を示す. 計算時間の計測には, NVIDIA RTX A6000 を使用した.	54
4.3	Atari 2600 における 100 エピソード間の最大/平均スコア. ここで, 各手法における 5 試行のうち, 平均スコアが最も高かったモデルを用いた. また, 太字は各ゲームにおける最大/平均スコアの最高スコアを示す.	59
4.4	Policy mask-attention の注視領域を反転させた場合とさせなかった場合のスコア比較. ここで, inverse attention の ✓ は, mask-attention の注視領域を反転させるか否かを示す. また random はランダムな行動選択によるスコアである.	59
4.5	ロボットマニピュレーションタスクにおける実験条件.	64
4.6	ロボットマニピュレーションタスクにおける 1,000 エピソード間の把持成功率	67
4.7	エージェントの行動予測に対する設問の平均正答率と unknown 回答率. w/o attention は RGB 画像のみの提示による教示方法である.	69
5.1	100 エピソード間におけるタスク達成率	86
5.2	ロボットの振る舞いに対する評価設問の平均正答率 [%]	88

第1章

序論

本章では、本研究の背景及び目的、本論文の構成について述べる。

1.1 研究の背景

深層強化学習とは、エージェントと環境間のインタラクションにもとづく経験から、環境におけるエージェントの最適な振る舞いを学習する技術である。この技術は、人間によるアノテーションデータから学習する教師あり学習と比べ、環境とのインタラクションを通じて獲得した経験から学習することで、人間を超えるような高い制御性能が獲得できることが知られている。この特徴から、ビデオゲームタスクやロボティクスをはじめとした動的な制御が求められる問題に対し応用が期待されている。例えば、囲碁 AI である AlphaGo/Zero [6, 7] や、単腕アームロボットによるマニピュレーション [8]、自動運転車 [9]、言語モデルのアライメント [10] などが挙げられる。これら様々なタスクで高い性能を発揮しているが、深層強化学習には 1) 学習効率の低さに起因するエージェントモデルの性能に対する課題と、2) 高性能なエージェントモデルの意思決定プロセスがブラックボックスであることに起因するエージェントモデルの信頼性に対する課題がある。

1) 深層強化学習における学習は、環境とのインタラクションを通じて学習データを収集するため、エージェントモデルの学習に大幅な計算資源と時間を要する。この課題に対して、マルチタスク学習の一種である補助学習の導入による学習効率向上が報告されている。補助学習とは、メインタスクとは異なる目標を持つタスク (補助タスク) を、メインタスクと同一モデルで学習することで、メインタスクの学習を促進させるアプローチである。この深層強化学習における補助学習の導入では、RGB 画像から深度情報を推定する補助タスク [11] や、エピソード終了を予測する補助タスク [12]、教師なし補助タスク [13] など、いくつかの補助タスクが提案されている。一方で、これら全ての補助タスクは必ずしもメインタスクに貢献するわけではなく、あらかじめメインタスクに適した補助タスクを人手で選定する必要がある。この人手による補助タスクの選定は、明確な基準が存在せず、メインタスクごとに有効な補助タスクが異なるため、どのような補助タスクがメインタスクに有効か判断が非常に困難である。そのため、これら補助タスクを動的に選択可能なアプローチが期待されている。

2) 深層強化学習エージェントの高性能化の背景には、深層学習の恩恵が大きく関わっており、説明可能な AI (eXplainable AI; XAI) 分野と同様に、エージェントモデル内部のパラメータ増加と演算処理の複雑化に伴い、エージェントモデルの判断根拠がブラックボックスとなっている。この課題は、エージェントモデルがなぜそのような行動を選択したか、エージェントモデルの判断根拠がユーザには理解できないことであり、深層強化学習の信頼性に直結する重大な課題である。このような強化学習における説明性や解釈性向上を目的とした研究分野は、XAI になぞらえて説明可能な強化学習 (eXplainable Reinforcement Learning; XRL) と呼ばれている。XRL 分野におけるアプローチとして、エージェントモデルに対する視覚的説明がある [4, 5]。この視覚的説明は、エージェントモデルの行動選択に対する判断根拠を顕著性マップや attention map など、画像情報として解釈可能とするアプローチである。このアプローチは、ユーザがエージェントモデルの意思決定を直感的に理解できるため、エージェントモデルの意思決定に対する視覚的説明の実現が期待されている。一方で、視覚的説明には最終的なエージェントモデルの意思決定に対する洞察が、ユーザの主観に依存してしまうという課題も存在する。そこで、エージェントモデルに対する言語的説明も研究されている

[14, 15]. この言語的説明は、エージェントモデルの行動選択に対する判断根拠を自然言語による文章情報として解釈可能にするアプローチである。言語的説明は、視覚的説明と異なり、エージェントモデルの意思決定に対する洞察を得る際にユーザの主観に依存しにくい。また、自然言語処理分野では大規模言語モデル [16, 17, 18] の登場により、大幅な技術進歩が起きていることから、視覚的説明と同様にエージェントモデルの意思決定に対する言語的説明の実現が期待されている。

1.2 研究目的

本研究では、以下の2つの項目を目的とする。

1. 深層強化学習エージェントモデルの性能向上.
2. 深層強化学習エージェントモデルの意思決定プロセスに対する説明性/解釈性の向上.

1つ目は、補助学習を導入した深層強化学習手法を対象とし、メインタスクを考慮した深層強化学習による補助タスクの動的な選択を提案する。2つ目は、深層強化学習エージェントモデルのブラックボックス問題に対し、方策と状態価値に着目した深層強化学習エージェントに対する視覚的説明手法、およびエージェントモデルが選択可能な行動全てに着目した深層強化学習エージェントに対する視覚的説明手法を提案する。以下に、2項目における本研究の目的について述べる。

深層強化学習エージェントモデルの性能向上 深層強化学習における学習は、環境とのインタラクションを通じて学習データを収集するため、エージェントモデルの学習に大幅な計算資源と時間を要する。この課題に対して、エージェントモデルの学習時に補助学習を導入するアプローチがある。しかしながら、補助学習に用いる補助タスクはメインタスクに適している必要があり、補助タスクを選定する際の明確な基準がないため、あらかじめ人手によって補助タスクを選定することは現実的ではない。またこの補助タスクの選定は、エージェントモデルの性能に直結するため非常にセンシティブである。このような理由から、本研究では深層強化学習による補助タスクの動的な選択を提案する。補助タスクの損失に対するバイナリ重みを深層強化学習により制御することで、補助タスクの選択を実現する。本手法は、メインタスクの学習段階に合わせた補助タスクの動的な選択を実現し、メインタスクに対するエージェントモデルの性能向上に貢献する。

深層強化学習エージェントモデルの意思決定プロセスに対する説明性/解釈性の向上 深層強化学習エージェントの高性能化の背景には、深層学習の恩恵が大きく関わっており、エージェントモデル内部のパラメータ増加と演算処理の複雑化に伴い、エージェントモデルの意思決定に対する判断根拠がブラックボックスとなっている。この課題は、エージェントモデルがなぜそのような行動を選択したか、エージェントモデルの判断根拠がユーザには理解できないことである。これは、深層強化学習エージェントに対する信頼性に直結する重大な課題である。この課題に対して、深層強化学習の枠組みにおける方策と状態価値に着目したエージェントモデルに対する視覚的説明を提案する。この手法は、Actor-Critic ベースな深層強化学習手法を対象とし、エージェントモデルの出力ブランチに

attention 機構を導入する。これにより、方策と状態価値の2つの観点からエージェントモデルの行動選択に対する判断根拠の視覚的な解析を可能とする。また、同様の目的に対し、選択可能な行動全てに着目したエージェントモデルの意思決定プロセスに対する視覚的説明も提案する。この手法では、行動情報を query とする Transformer encoder-decoder 構造にもとづきエージェントモデルを構築する。これにより、decoder 部で行動ごとにエージェントモデルの注視領域を示す固有の attention weight を獲得可能とすることで、エージェントモデルの意思決定に対する行動ごとの網羅的な視覚的説明を実現する。

本研究の目的は、深層強化学習エージェントの意思決定プロセスに対する説明性/解釈性の向上である。エージェントモデルに対する視覚的説明手法の多くは、エージェントモデルの注視領域を入力画像に重畳した顕著性マップや attention map など、画像情報としてユーザに解釈を提供している。しかし、ロボット制御のような実空間で動作するエージェントモデルにおいて、画像情報による視覚的フィードバックでは奥行きなどの3次元情報が欠落し、ユーザの直感的な理解との結びつきが弱い。この課題に対し、エージェントモデルの判断根拠に対する Augmented Reality (AR) を用いた視覚的フィードバックを提案する。エージェントモデルの判断根拠を AR を介してユーザに提供することで、3次元情報を考慮したエージェントモデルの意思決定に対する洞察の提供を可能とする。また、深層強化学習エージェントに対する視覚的説明では、エージェントモデルの判断根拠に対する最終的な洞察はユーザの主観に依存してしまっている。そのため、さらなる説明性/解釈性向上にむけて、エージェントモデルに対する自然言語にもとづく言語的説明が期待されている。このような理由から、視覚的説明情報を活用した大規模言語モデル (LLM) による言語的説明手法を提案し、視覚的説明から言語的説明への拡張について議論する。言語的説明へ拡張することで、エージェントモデルの意思決定に対する洞察をより直感的かつ明確な説明が期待できる。

1.3 本論文の構成

本論文は、図 1.1 に示すように7つの章で構成されている。1章では、本研究の背景と目的を述べた。本研究では、深層強化学習における補助学習を用いたエージェントモデルの性能向上、方策と状態価値および、選択可能な全ての行動に着目したエージェントモデルに対する視覚的説明、エージェントモデルに対する言語的説明、それぞれに対する枠組みを提案する。

2章では、基礎技術となる深層強化学習について、その応用事例と課題、および説明可能な強化学習の研究動向について述べる。深層強化学習アルゴリズムについて体系的にまとめ、それら深層強化学習の応用事例と課題を紹介する。深層強化学習の代表的な課題の一つであるエージェントモデルの信頼性に焦点を当て、その研究分野である説明可能な強化学習についての関連研究を体系的にまとめる。3章では、深層強化学習における性能向上のアプローチである補助学習に着目し、深層強化学習エージェントモデルのメインタスクに合わせた補助タスクの動的な選択について述べる。4章では、エージェントモデルの説明性/解釈性向上を目的とし、方策と状態価値に着目した深層強化学習エージェントモデルの判断根拠に対する視覚的説明手法について述べる。5章では、4章と同様の目的のもと、エージェントモデルが選択可能な行動全てに着目した深層強化学習エージェントモデル

の判断根拠に対する視覚的説明手法について述べる。また、エージェントモデルの意思決定プロセスに対する視覚的説明情報を、ユーザへ直感的かつ即時的に提供可能とする Augmented Reality (AR) を用いた視覚的フィードバック手法についても述べる。そして、6章では4章と5章で得た視覚的説明情報を利用し、エージェントモデルの判断根拠に対して、ユーザの直感的な理解が可能な自然言語による言語的説明への拡張について述べる。7章では、本論文の結論と展望について述べる。

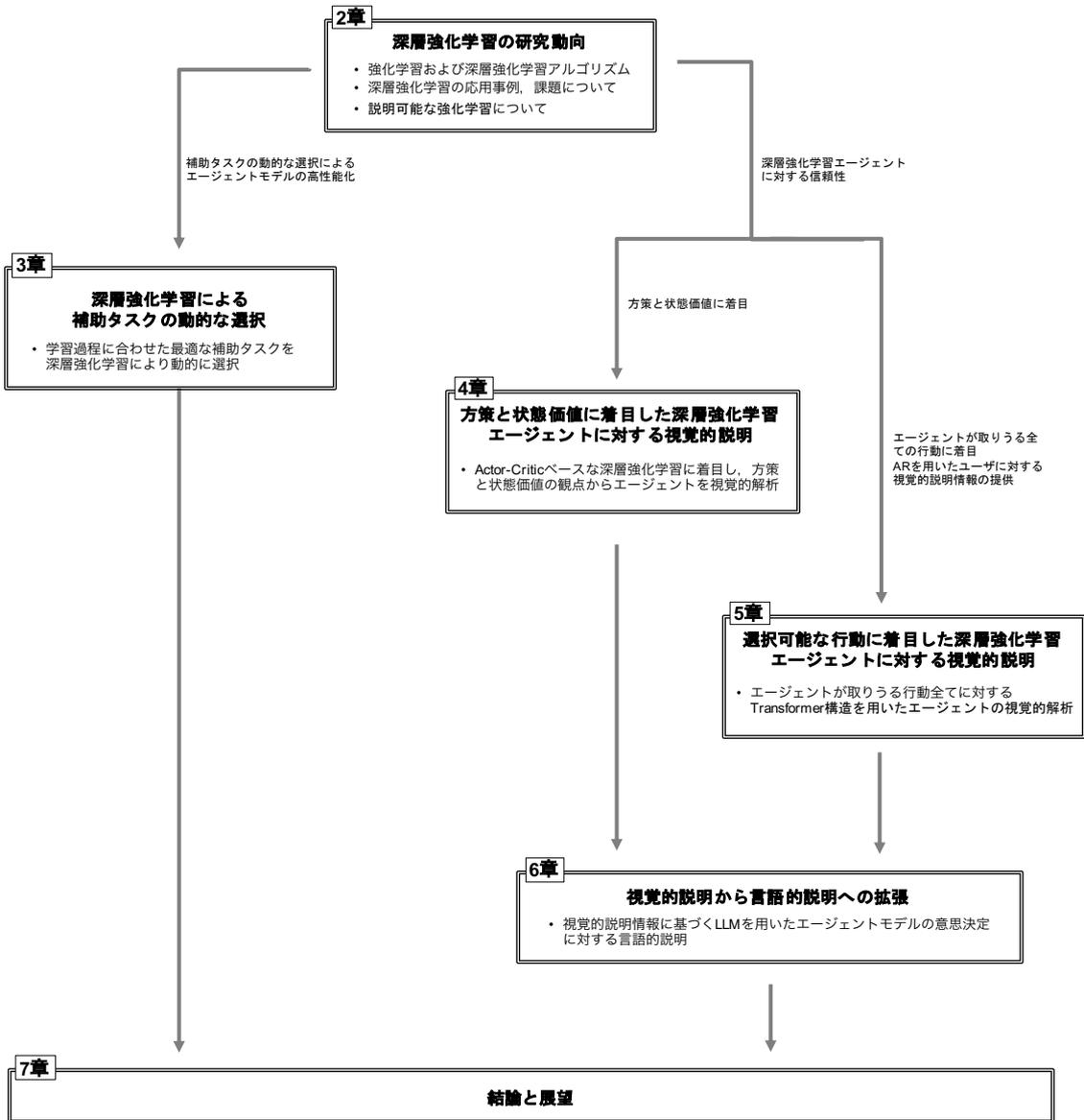


図 1.1: 本論文の構成.

第2章

深層強化学習の研究動向

強化学習とは、エージェントが環境上で試行錯誤を繰り返しながら、報酬につながる最適な振る舞いを学習する技術である。

本章の構成は以下のとおりである。まず、2.1 節では強化学習を始めとし深層強化学習アルゴリズムの研究動向について述べる。2.2 節では、2.1 節で述べた深層強化学習アルゴリズムが応用されている事例について述べる。2.4 節では、2.2 節で述べた応用事例の課題として挙げられる深層強化学習エージェントの信頼性に関する研究動向について述べる。

2.1 強化学習

本節では、はじめに強化学習の基礎について説明する。次に、数多く提案されている強化学習アルゴリズムの分類方法について触れる。そして、方策の表現方法による分類に着目し、Value ベースと Policy ベース、Actor-Critic ベースの代表的な手法について述べる。また、自然言語処理分野から始まり、画像認識分野など多岐に渡って大きなブレイクスルーとなった Transformer [2] に着目し、この構造を導入した深層強化学習アルゴリズムについて述べる。

強化学習 (Reinforcement Learning; RL) [19] とは、機械学習の一種であり、ある問題設定において試行錯誤を繰り返しながら、報酬につながる最適な行動を獲得する学習方法である。他の機械学習手法とは異なり、どのような行動を選択すれば良いかは与えられず、どの行動を選択すればより良い結果に結び付くかを見つけ出す学習方法である。強化学習における学習データでは、「データセット」ではなく「環境」から観測することで取得するため、教師あり学習のようにラベル付きデータセット等を作成する必要がない。そのため、ビデオゲーム攻略やロボット制御など、教師データの作成が困難なタスクへの応用が期待されている [20, 21, 22, 23]。また、強化学習では環境の開始から終了までの期間を「1 エピソード」、1 エピソード間における報酬の総和を「収益」と定義する。強化学習は、この収益を最大化するように最適化対象の振る舞いを学習することが目的である。

■ マルコフ決定過程

強化学習における環境は、マルコフ性 (Markov property) を持っていることを想定している。マルコフ性とは、「遷移先の状態 s' は遷移前の状態 s とそこでの行動 a のみに依存する。報酬は遷移前の状態 s と遷移先に依存する。」という性質のことである。また、マルコフ性を持った環境をマルコフ

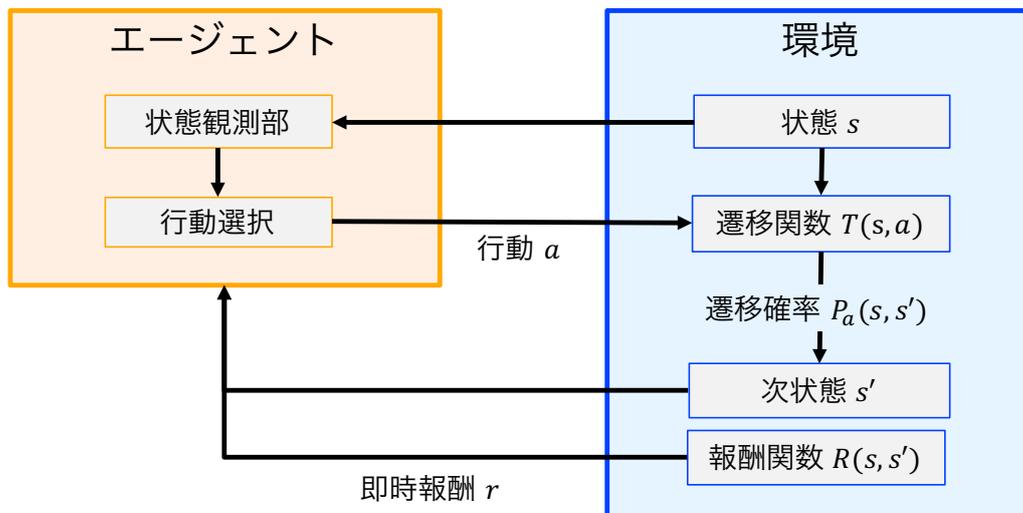


図 2.1: 強化学習の概要

決定過程 (Markov Decision Process; MDP) と呼ぶ。MDP の構成要素は以下の 4 つである。

- 状態 (State): s
- 行動 (Action): a
- 遷移関数 (Transition function): T
状態と行動から、次状態と遷移確率を出力する関数.
- 報酬関数 (Reward function): R
状態と次状態から、報酬を出力する関数.

環境とは異なり、状態から行動を選択する関数を「戦略/方策 (Policy)」, 方策に従い動く主体を「エージェント (Agent)」と呼ぶ。この方策のパラメータを、エージェントと環境のインタラクションを通して、状態に対し適切な行動を取るようチューニングすることが、強化学習における学習である。また MDP における報酬は、「遷移前の状態 s と遷移先の状態 s' に依存する」という性質を持つため、即時報酬 (Immediate reward) と呼ぶ。したがって、MDP における収益とは即時報酬の合計を意味する。これらを踏まえた強化学習の概要を図 2.1 に示す。

強化学習における基本的なサイクルを以下で述べる。

1. エージェントが環境の状態 s を観測.
2. エージェントが観測した状態 s から行動 a を選択.
3. 環境の状態が遷移関数 $T(s, a)$ にもとづき次状態 s' に遷移.
4. 状態 s と次状態 s' から報酬関数 $R(s, s')$ にもとづき即時報酬 r を算出し、エージェントへフィードバック.
5. 1 へ戻る.

上記のサイクルを繰り返し、行動選択 (方策) を更新することでエージェントの最適な振る舞いの獲得を図る。

■ 価値

強化学習において最適な行動を学習するには、選択した行動が良い行動であるか悪い行動であるか評価する必要がある。最適な行動とは、より大きな収益が得られる行動のことである。収益 G_t は 1 エピソード間の即時報酬の総和であるため、式 (2.1) のように表すことができる。

$$G_t \stackrel{\text{def}}{=} r_{t+1} + \gamma G_{t+1} \quad (2.1)$$

ここで、 γ は未来の即時報酬が不確かな値であることを考慮するための割引率である。式 (2.1) に示した収益 G_t を、期待報酬 (Expected reward) または価値 (Value) と呼ぶ。強化学習ではこの価値を用いることで行動に対する評価を行う。しかし、式 (2.1) で表した価値では、未来の即時報酬が既知であり、必ず得られなければ算出できないという問題がある。本来、環境から得られる即時報酬はエージェントが環境上で実際に行動してみない限り知ることができない。そこで、即時報酬に確率を乗算し、期待値として価値を算出することで、この問題を解決する。

行動確率を定義できれば、行動の結果から得られる報酬 (即時報酬) に行動確率を乗算することで、期待値を算出することができる。強化学習における行動には 2 つの定義がある。1 つ目は「エージェントが保持する方策にもとづき行動を選択」、2 つ目は「エージェントは常に価値が最大となる行動を選択」である。1 つ目の定義から、ある状態 s から方策 π にもとづいて行動することで得られる価値 $V_\pi(s)$ を式 (2.2) のように表すことができる。

$$V_\pi(s) = E_\pi[r_{t+1} + \gamma V_\pi(s')] = \sum_a \pi(a|s) \sum_{s'} T(s'|s, a)(R(s, s') + \gamma V_\pi(s')) \quad (2.2)$$

ここで、 $\pi(a|s)$ は状態 s において行動 a を選択する確率、 $T(s'|s, a)$ は状態 s において行動 a を行った際に状態 s' へ遷移する確率、 $R(s, s')$ は状態 s から状態 s' へ遷移した際の報酬である。式 (2.2) に示した $V_\pi(s)$ を状態価値 (State value) と呼ぶ。このように、価値を算出する式をベルマン方程式 (Bellman equation) と呼ぶ。2 つ目の定義である「価値が最大となる行動を選択」の場合も、ベルマン方程式により式 (2.2) と同様に表すことができる。2 つ目の定義における状態価値 $V_\pi(s)$ を式 (2.3) に示す。

$$V_\pi(s) = \max_a \sum_{s'} T(s'|s, a)(R(s, s') + \gamma V_\pi(s')) \quad (2.3)$$

式 (2.3) は、式 (2.2) の $\sum_a \pi(a|s)$ を \max_a に置き換えている。

また、状態価値 $V_\pi(s)$ に対し、ある状態 s においてある行動 a を選択し、その後方策 π に従うことで得られる価値を行動価値 $Q^\pi(s, a)$ と呼ぶ。行動価値 $Q^\pi(s, a)$ を式 (2.4) に示す。

$$Q^\pi(s, a) = \sum_{s'} T(s'|s, a)(R(s, s') + \gamma V_\pi(s')) \quad (2.4)$$

行動価値 $Q^\pi(s, a)$ は行動 a 後の価値であるため、式 (2.4) は式 (2.2) から行動に寄与する箇所を取り除いただけである。

■ TD 法

強化学習において、価値が正しく推定できることは最適な行動を獲得したことに繋がる。あらゆる状態において状態価値 $V(s)$ が正しく推定できている場合、最も状態価値が高くなる行動を常に選択することが、環境でエージェントが取るべき最適な行動となる。(行動価値 $Q(s, a)$ は行動 a 後の価値であるため、最も行動価値の高い行動を選択すれば良い。)

価値を正しく推定するための手法の一つに、TD 法 (Temporal Difference learning) がある。TD 法とは、推定の価値と実際に行動し得られる価値の差が 0 となるように価値の推定を修正する方法である。推定の価値と実際に行動し得られる価値との差を TD 誤差 δ と呼び、式 (2.5) のように表すことができる。

$$\delta = r + \gamma V(s') - V(s) \quad (2.5)$$

ここで、 $r + \gamma V(s')$ が実際に行動し得られた価値を表し、 $V(s)$ が推定の価値である。TD 誤差が大きい場合は価値の推定が正しく行えておらず、小さい場合は正しく推定が行えていることを意味する。

TD 誤差を用いた価値の更新は、行動後に行動前の価値に対して行う。TD 誤差を用いた価値の更新を式 (2.6) に示す。

$$V(s) \leftarrow V(s) + \alpha \cdot \delta \quad (2.6)$$

ここで、 α は学習を制御するパラメータで学習率と呼ぶ。 α が 1 の場合は、推定の価値 $V(s)$ が実際の価値 $r + \gamma V(s')$ に置き換えられる。

2.1.1 強化学習アルゴリズムの分類

強化学習アルゴリズムは、何を基準とするかでいくつかのカテゴリが存在する。主なカテゴリとして「環境モデルの有無による分類」、「方策改善時の行動による分類」、「学習データの収集方法による分類」「方策の表現方法による分類」が挙げられる。以下でそれぞれの分類方法について述べる。

環境モデルの有無による分類。 強化学習の枠組みにおける環境をモデル化し、エージェントの方策改善に利用するかどうかで分類するカテゴリである。環境から得た経験を用いて環境モデルを学習し、方策改善に利用する強化学習アルゴリズムを Model-base、環境モデルを持たず、環境から得た経験を使って方策改善する強化学習アルゴリズムを Model-free と呼ぶ。

方策改善時の行動による分類。 方策改善に用いる行動を実際の学習データ収集時の方策とするかどうかで分類するカテゴリである。学習データ収集時に選択した行動にもとづいて方策改善

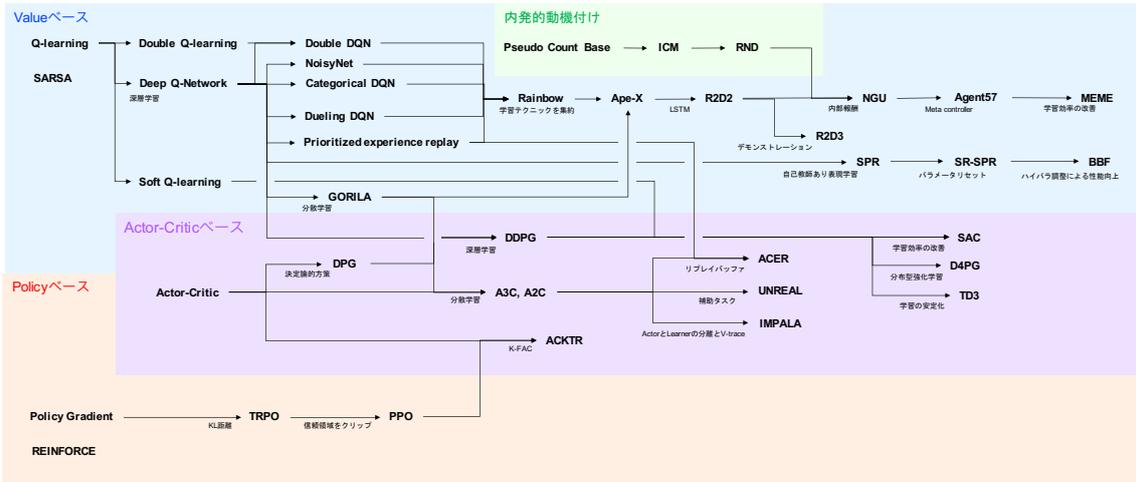


図 2.2: 強化学習のアルゴリズムマップ (方策の表現方法による分類)

する強化学習手法を On-policy, 現方策が最善と捉える行動にもとづいて方策改善する強化学習手法を Off-policy と呼ぶ。

学習データの収集方法による分類. エージェントの方策改善に用いる学習データをどのように収集するかで分類するカテゴリーズである。エージェントが環境と直接インタラクションすることで学習データを収集する強化学習手法を Online RL, あらかじめ収集された学習データをもとに方策改善する強化学習手法を Offline RL と呼ぶ。

方策の表現方法による分類. 行動選択の基準を何にするかで分類するカテゴリーズである。エージェントの方策を条件付き確率と定義し, 方策勾配法により学習する強化学習手法を Policy ベース, エージェントの方策を行動価値関数と定義し, TD 学習によって価値の更新を行う強化学習手法を Value ベースと呼ぶ。またこれら 2 つの手法を組み合わせた Actor-Critic ベースも存在する。

方策の表現方法による分類は, 強化学習エージェントモデルの振る舞いに大きく影響する代表的な分類方法である。以下では, この分類方法に焦点を当て, 代表的な深層強化学習アルゴリズムについて述べる。Value ベースは価値が最大となる状態に遷移するように行動を決定し, Policy ベースは方策にもとづいて行動を決定する。図 2.2 に Value ベースと Policy ベース, Actor-Critic ベースに分類した強化学習のアルゴリズムマップを示す。

Value ベースは, TD 誤差を 0 に近づけるように行動価値 $Q(s, a)$ の推定を修正し, 最適な行動価値を求める。Value ベースの手法は, 行動選択に価値関数を用いて行動を決定するため, 離散値制御タスクにおいて広く研究されている [24, 25, 26, 27, 28, 29, 30, 31, 1, 32, 33, 34, 35, 36, 37]。

Policy ベースは, 現在の方策を改善することを目的としており, 方策 π_θ のパラメータ θ を勾配に沿って更新することで最適な方策を求める。勾配の算出には, ある方策 π_θ の収益の期待値を表す目的関数 E^{π_θ} をパラメータ θ で微分することで求めている。Policy ベースの手法は, 行動選択に方策を用いるため連続値制御タスクにおいて広く研究されている [38, 39, 40, 41]

また, 経験を価値と方策どちらの更新にも用いる Actro-Critic ベースがある。価値と方策どちらの更新にも用いているため, Value ベースと Policy ベースを組み合わせた手法であると考えられ

る。Value ベースと Policy ベースどちらのメリットも持っていることから、盛んに研究されている [42, 43, 44, 45, 46, 13, 47, 48, 49, 50]。まずはじめに、Value ベースと Policy ベース、Actor-Critic ベースそれぞれにおける基礎的な手法について述べる。Value ベースは Q-learning [24]、Policy ベースは Policy gradient [39]、Actor-Critic ベースは Actor-Critic 法 [42] について説明する。

■ Q-learning [24]

Q-learning (Q 学習) とは、TD 法を用いた代表的な手法の 1 つで、正確な行動価値 (Q 値や Q 関数と呼ばれる) を推定するように学習を行う手法である。Q 学習では、全ての状態 s と行動 a に対し行動価値 $Q(s, a)$ のテーブル (Q-table) を作成する。はじめに、Q-table を任意の値に初期化する。この初期段階では、各状態と行動に対する正確な行動価値は分からない。そこで、あるゆる状態の下で多様な行動を選択し、その時に獲得した報酬を用いて推定した行動価値を修正する。この処理により、正確な行動価値を表したテーブルとなるように Q-table を更新する。この時の多様な行動選択には、 ϵ -greedy 法が用いられる。 ϵ -greedy 法とは、ある状態において一定確率 ϵ で取り得る行動の中からランダムに行動選択し、 $1 - \epsilon$ の確率で行動価値が最も高い行動を選択する方法である。行動価値 $Q(s, a)$ の更新を式 (2.7) に示す。

$$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a)) \quad (2.7)$$

式 (2.7) で、 s は状態、 s' は次状態、 a は行動、 r は報酬を表す。 $\max_{a'} Q(s', a')$ は、次状態 s' における最も高い行動価値である。 γ ($0 < \gamma \leq 1$) は割引率であり、計算には 0.9~0.99 が使用される。 α ($0 < \alpha \leq 1$) は学習係数であり、計算には 0.1 程度がよく使用される。式 (2.7) は、行動価値 $Q(s, a)$ よりも、次状態 s' における最良の行動価値 $Q(s', a')$ の方が大きければ $Q(s, a)$ を大きくし、逆に小さければ $Q(s, a)$ を小さくすることを表す。つまり、ある状態におけるある行動の価値を次状態における最良の行動価値に近似する。

Q 学習では、学習データ収集時において ϵ -greedy 法による行動選択が採用され、価値更新には現行動価値において最良と推定している行動価値を用いる (式 (2.7) の $\max_{a'} Q(s', a')$)。一方で、価値更新にも ϵ -greedy 法の概念を導入した手法を State-Action-Reward-State-Action (SARSA) と呼ぶ。つまり、Q 学習は Value ベースかつ Off-policy な手法であり、SARSA は Value ベースかつ On-policy な手法である。

■ Policy gradient [39]

Policy gradient (方策勾配法) とは、パラメータを持つ関数で方策を表現し、このパラメータを勾配法で最適化する手法である。この手法は Policy ベースの基礎に位置する手法である。方策のパラメータ更新は、2.1 節の TD 法とは異なり、算出した価値と直接比較することができない。そこで、式 (2.2) と同様にベルマン方程式を用いて価値の期待値 (方策の価値) を考える。これは計算式上で見ると、確率が 1 つ増えるだけで、「(方策に従い) 状態に遷移する確率」と「行動確率」、「行動価値」か

ら期待値 $J(\theta)$ を算出することができる。期待値 $J(\theta)$ を式 (2.8) に示す。

$$J(\theta) \propto \sum_{s \in S} d^{\pi_\theta}(s) \sum_{a \in A} \pi_\theta(a|s) Q^{\pi_\theta}(s, a) \quad (2.8)$$

ここで、 $d^{\pi_\theta}(s)$ が方策 π_θ に従い状態 s へ遷移する確率、 $\pi_\theta(a|s)$ が行動確率、 $Q^{\pi_\theta}(s, a)$ が行動価値を表している。行動確率と行動価値から状態価値を算出 ($\sum_{a \in A} \pi_\theta(a|s) Q^{\pi_\theta}(s, a)$) し、状態価値に遷移確率を掛け合算することで、期待値を計算している。また、 \propto は比例関係を表す記号である。

式 (2.8) に示した期待値 $J(\theta)$ を最大化するために、勾配法を用いてパラメータ θ の修正を行う。期待値の勾配 $\nabla J(\theta)$ を式 (2.9) に示す。

$$\nabla J(\theta) \propto \sum_{s \in S} d^{\pi_\theta}(s) \sum_{a \in A} \nabla \pi_\theta(a|s) Q^{\pi_\theta}(s, a) \quad (2.9)$$

式 (2.9) の導出に必要な定理を、方策勾配定理 (Policy gradient theorem) と呼ぶ。対数の微分の定義より $\nabla \pi_\theta(a|s) = \pi_\theta(a|s) \frac{\nabla \pi_\theta(a|s)}{\pi_\theta(a|s)} = \pi_\theta(a|s) \nabla \log \pi_\theta(a|s)$ と変形できる。式 (2.9) から $\nabla \pi_\theta(a|s)$ を変形すると式 (2.10) となる。

$$\nabla J(\theta) \propto \sum_{s \in S} d^{\pi_\theta}(s) \sum_{a \in A} \pi_\theta(a|s) \nabla \log \pi_\theta(a|s) Q^{\pi_\theta}(s, a) \quad (2.10)$$

ここで、 $d^{\pi_\theta}(s), \pi_\theta(a|s)$ が確率を表している。また、期待値の形式に式 (2.10) を変形すると式 (2.11) となる。

$$\nabla J(\theta) \propto E_{\pi_\theta}[\nabla \log \pi_\theta(a|s) Q^{\pi_\theta}(s, a)] \quad (2.11)$$

$\nabla \log \pi_\theta(a|s) Q^{\pi_\theta}(s, a)$ は、勾配である $\nabla \log \pi_\theta(a|s)$ が移動方向、行動価値である $Q^{\pi_\theta}(s, a)$ がその度合いと解釈することができる。

■ Actor-Critic 法 [42]

Actor-Critic 法とは、行動選択と状態評価を actor と critic として、それぞれ独立させ同時に更新し学習する手法である。また、それぞれが独立していることで行動選択に必要な計算コストが少ない点と、確率的な方策を学習できる点の 2 つの利点がある。図 2.3 に Actor-Critic 法の構成を示す。Actor-Critic 法において、エージェントは actor と critic から構成される。Actor は、ある状態において行動を選択する確率分布で、確率の方策 (stochastic policy) を持ち行動を選択する。Critic は、actor が出力する確率の方策に対する状態の評価値 (状態価値) を推定する。Critic が推定する状態価値を用いて、式 (2.5) により TD 誤差を算出する。この TD 誤差を用いて、actor と critic を同時に更新する。ここで TD 誤差を用いた Actor の更新は、TD 誤差 > 0 ならば選択した行動が予測より良かったことを意味し、その行動の確率を上げる。また、TD 誤差 < 0 ならば選択した行動が予測より悪かったこ

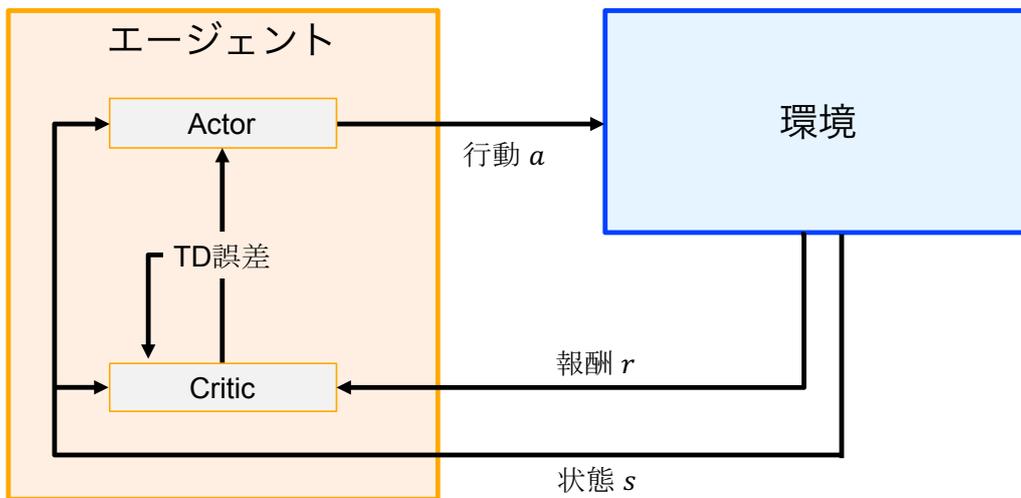


図 2.3: Actor-Critic 法の構成

とを意味し、その行動の確率を下げることである。TD 誤差を方策と価値の更新に用いているため、Actor-Critic 法は policy ベースと value ベースどちらの要素も兼ね備えていると言える。

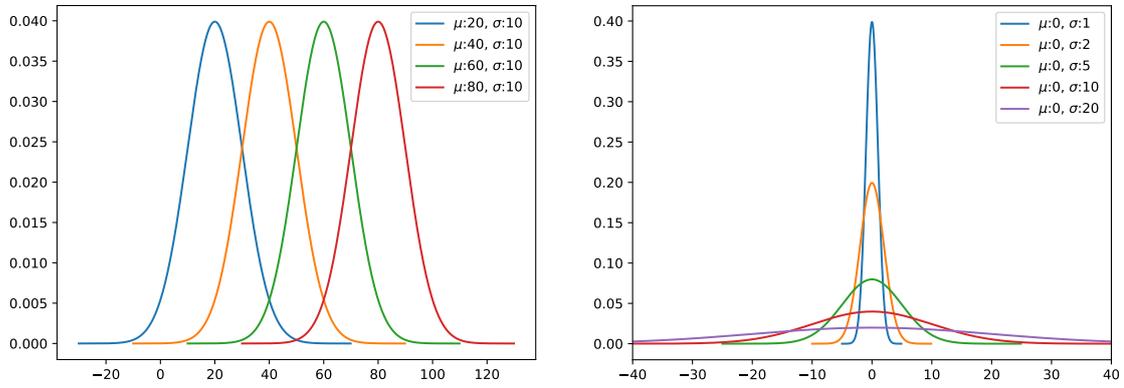
Actor-Critic 法における actor の確率的方策は、一般的に行動空間が離散であること想定している。しかし、actor の方策を確率密度関数とすることで、連続値の行動にも適用することができる。行動空間が連続である場合、actor の確率的方策は入力である状態に応じて、平均値と分散が変化する正規分布と定義する。図 2.4 (a) に平均値 μ を変動させた場合の確率密度関数の変化、図 2.4 (b) に標準偏差 σ を変動させた場合の確率密度関数の変化を示す。図 2.4 から、平均値 μ と標準偏差 σ を変動させることで、確率分布を様々な形にできることが分かる。行動空間が連続である場合の方策 $\pi(a|s, \mu(s), \sigma(s))$ を式 (2.12) に示す。

$$\pi(a|s, \mu(s), \sigma(s)) = \frac{1}{\sigma(s)\sqrt{2\Pi}} \exp\left(-\frac{(a - \mu(s))^2}{2\sigma(s)^2}\right) \quad (2.12)$$

ここで、 Π は円周率を表す。行動空間が連続である場合の Actor は、状態 s を観測し平均値 $\mu(s)$ と標準偏差 $\sigma(s)$ を出力する。そして、式 (2.12) に示すように平均値 $\mu(s)$ と標準偏差 $\sigma(s)$ に従う正規分布から行動 a を選択する。したがって、状態 s における最適な行動 a が最も確率が高くなるように、出力である平均値 $\mu(s)$ と標準偏差 $\sigma(s)$ を調整するように学習する。

2.1.2 Value ベースにおける代表的な深層強化学習手法

Value ベースな深層強化学習とは、方策の学習方法を Q-learning ベースとした深層強化学習アルゴリズムである。本節では、Value ベースの代表的な深層強化学習手法として、Deep Q-Network (DQN) [25], Rainbow [1] について述べる。



(a) 平均値 μ の変動

(b) 標準偏差 σ の変動

図 2.4: 確率密度関数

■ Deep Q-Network [25]

Deep Q-Network (DQN) とは、Q 学習と Deep Convolutional Neural Network (DCNN) を組み合わせた手法である。Q 学習では Q-table を用いて行動価値を表現していたが、DQN ではニューラルネットワークを用いた演算により行動価値を近似している。図 2.5 に DQN のネットワーク構造を示す。ネットワークの出力が最適な行動価値と同様になるように学習、すなわちネットワークのパラメータを更新する。誤差関数 L_θ を式 (2.13) に示す。

$$L_\theta = E \left[\frac{1}{2} (r + \gamma \max_{a'} Q_{\theta_i}(s', a') - Q_{\theta_i}(s, a))^2 \right] \quad (2.13)$$

ここで、 θ はネットワークの重みである。式 (2.13) において、 $r + \gamma \max_a Q_{\theta_i}(s', a')$ は教師あり学習での教師データにあたり、target と呼ぶ。また、誤差逆伝播する際の勾配は式 (2.13) を微分することで得られる。勾配 $\nabla L(\theta_i)$ を式 (2.14) に示す。

$$\nabla L(\theta_i) = E \left[(r + \gamma \max_{a'} Q_{\theta_i}(s', a') - Q_{\theta_i}(s, a)) \nabla Q_{\theta_i}(s, a) \right] \quad (2.14)$$

DQN では Experience replay, Target Q-network, Clipping reward と呼ばれる工夫を用いることで、Atari2600 におけるゲームタスクにおいて高スコアを獲得している。以下に、これらの工夫について説明する。

- Experience replay

環境とのインタラクションによって獲得した経験をメモリに蓄積し、学習の際はそのメモリから経験をランダムサンプリングする工夫である。ここでの経験とは、状態と行動、報酬、次状態を 1 セットにしたものである。強化学習では、経験は連続的であり、獲得した順序により相関が生じる。そのため、Experience replay を用いることで、相関を解消することで学習を安定させている。

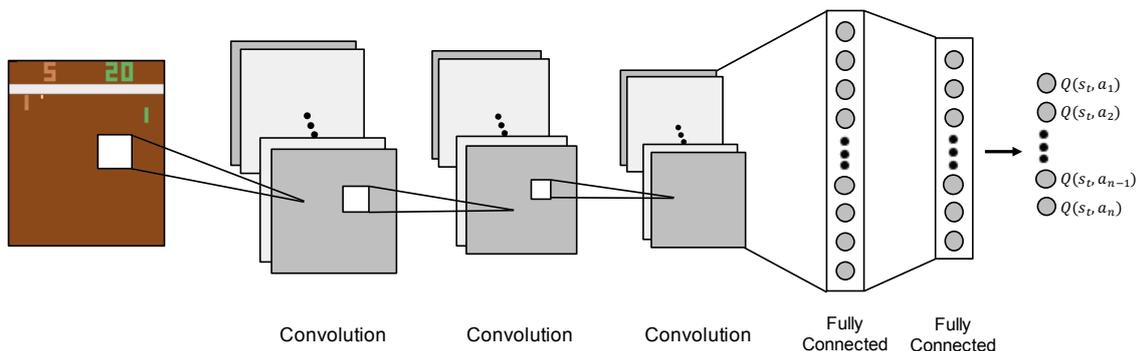


図 2.5: DQN のネットワーク構造

- Target Q-network

誤差関数 L_θ と勾配 $\nabla L(\theta_i)$ において, target を逐次的に更新するのではなく, 数ステップ毎に更新する工夫である. DQN における target は, 前パラメータ θ_{i-1} を用いて計算する. しかし, 学習毎に θ_{i-1} を更新すると, target が学習毎に変わるため, target への近似が困難になるという問題点が存在する. そのため, Target Q-network を用いることで, この問題を解決している.

- Clipping reward

報酬値のスケールを統一する工夫である. 強化学習では報酬を頼りに最適な行動を学習するため, 問題設定毎に報酬を設計する必要がある. このように報酬設計は問題設定に依存するため, スケールに大きな差が出る. そのため, 問題設定毎に最適なハイパーパラメータが異なり, 報酬設計によっては収束が困難になる場合がある. そこで, 正の報酬は +1, 報酬なしは 0, 負の報酬は -1 に統一する. この Clipping reward により, 様々な問題設定間で学習しやすくしている.

この手法は, Atari2600 において, 人間と同等かそれ以上のスコアを獲得している. DQN の登場以降, 強化学習に深層学習を組み込んだ深層強化学習が主流になった.

■ Rainbow [1]

2013 年に DQN が提案されてから, DQN に対し改良が施された手法が数多く提案されている. Rainbow は, DQN における有名な 6 つの派生手法を全て組み込んだ手法である. 価値の推定をより正しく行うための要素として, Double DQN と Dueling Network, Multi-step Learning, Distributional RL の 4 つが組み込まれている. 学習や探索の効率を改善するための要素として, Prioritized experience replay と Noisy Net の 2 つが組み込まれている. 以下に Rainbow に組み込まれた 6 つの要素について説明する.

- Double DQN [26]

DQN の行動選択では, 式 (2.13) に示すように, 行動価値が最も高い行動 ($\max_{a'} Q_{\theta_i}(s', a')$) を

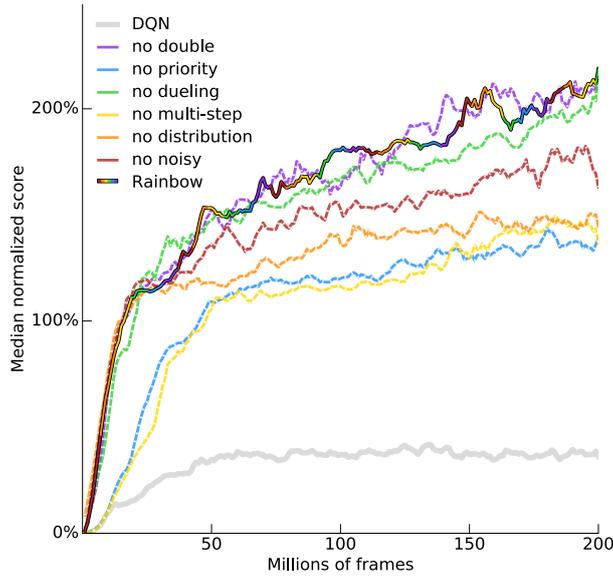


図 2.6: Rainbow における 6 つの要素の影響 (文献 [1] から引用)

選択している。しかし、行動価値の評価には誤差が含まれるため、本来、価値の低い行動を高く評価してしまうなど過大評価が起きている可能性がある。そこで、target を算出する際に、現ネットワークで行動を評価するのではなく、別のネットワークを用いる。つまり、2つのネットワークを構築し、1つ目を行動選択、2つ目を行動を評価するネットワークとして分離させる。この工夫を、Double DQN と呼ぶ。式 (2.15) に Double DQN での誤差関数 L_θ を示す。

$$L_\theta = E \left[\frac{1}{2} \left(r + \gamma Q_{\theta'_i} \left(s', \arg \max_{a'} Q_\theta(s', a) \right) - Q_{\theta_i}(s, a) \right)^2 \right] \quad (2.15)$$

式 (2.15) で、 $\arg \max_{a'} Q_\theta(s', a)$ が1つ目のネットワークによる行動である。したがって、 $Q_{\theta'_i}(s', \arg \max_{a'} Q_\theta(s', a))$ は1つ目のネットワークによる行動と、同じ行動を選択した場合の2つ目のネットワークの出力である。Double DQN の導入により、行動価値の過大評価を抑制している。

- Prioritized experience replay [29]

経験には、学習を促進させる経験とそうではない経験が存在する。DQN の Experience replay では、集めた経験からランダムサンプリングしているため、学習を促進させない経験も多くサンプリングされている。そこで、経験を溜め込んでいるメモリ内で、TD 誤差にもとづく経験の優先順位付けを行う。そして、この優先順位にもとづいて経験のサンプリングを行う。この手法を、Prioritized experience replay と呼ぶ。Prioritized experience replay の導入により、学習の効率を上げることができる。

- Dueling Network [27]

状態自体の価値 $V(s)$ と、状態における行動の価値 $Q(s, a)$ を分けて算出する。そして、行動価

値から状態価値を差し引いて、行動のみに対する価値を算出する。この時の行動のみに対する価値を Advantage と呼ぶ。この Advantage に関する詳細な説明は、2.1.4 節にて行う。Dueling Network を導入することで、算出する価値をより正確に推定することができる。

- Multi-step Learning [19]

DQN では 1 ステップの報酬を用いて target を算出している。この報酬を n ステップに拡張することで学習の効率化を図ることができる。ここで n ステップの報酬を考慮した TD 誤差 δ を式 (2.16) に示す。

$$\delta = r_{t+1} + r_{t+2} + r_{t+3} + \dots + \gamma^n \max_a Q(s_{t+n} - Q(s_t, a_t)) \quad (2.16)$$

- Distributional RL [30]

DQN では行動価値を収益の期待値として表現している。Distributional RL は、この期待収益を分布として扱うことで、その平均や分散は状態や行動によって変化するものであると定義する。これにより、分散のパラメータを使用することで、期待収益は同じであっても報酬のばらつきがあるケースを表現できる。このように、Distributional RL を用いることで、報酬に対する表現力を上げることができる。

- Noisy Nets [28]

DQN では探索を行うために、 ϵ -greedy 法を用いている。しかし、探索と利用はトレードオフの関係であり、 ϵ の設定に明確な値がないという問題がある。そこで、 ϵ もニューラルネットにより学習させることで、この問題を解決する。詳細としては、ネットワーク構造における全結合層を式 (2.17) のように定義する。

$$y = (W + \sigma^w \odot \epsilon^w)x + (b + \sigma^b \odot \epsilon^b) \quad (2.17)$$

ここで、 ϵ はランダムなノイズ、 σ はノイズを調整する値である。全結合層の重み W とバイアス b にノイズを入れる。つまり、どれくらいランダムな行動をした方が良いかを学習することである。この工夫を Noisy Nets と呼ぶ。Noisy Nets を導入することで、探索の効率を改善している。

図 2.6 に、Rainbow における 6 つの要素の影響を表したグラフを示す。ここで、Rainbow から各要素を除いた場合、スコアのグラフがどのように変化するかを示している。図 2.6 から、Rainbow において Prioritized experience replay と Multi-step Learning が最も重要であると言える。しかし、この結果は Atari2600 の様々なゲーム全体における平均であるため、どの要素が効くかはタスクに依存する。

2.1.3 Policy ベースにおける代表的な深層強化学習手法

Policy ベースな深層強化学習とは、方策の学習方法を方策勾配法ベースとした深層強化学習アルゴリズムである。本節では、Policy ベースの代表的な深層強化学習手法として、Trust Region Policy

Optimization (TRPO) [40], Proximal Policy Optimization (PPO) [41] について述べる.

■ Trust Region Policy Optimization [40]

方策勾配法 [39] には, 学習結果が安定しないという問題がある. この問題に対し改善を図った手法として, Trust Region Policy Optimization (TRPO) がある. TRPO では, パラメータ更新前の方策から更新後の方策が, 大幅に変化しないように KL ダイバージェンスを用いた制約をかけることで, 学習の安定化を図っている. パラメータ θ を持つ方策 $\pi_\theta(a_t|s_t)$ における方策勾配 \hat{g} は, 式 (2.18) のように表される.

$$\hat{g} = E_t[\nabla_\theta \log \pi_\theta(a_t|s_t) A_t] \quad (2.18)$$

ここで, A_t は Advantage を表している. 式 (2.18) から, TRPO における最大化を図る期待値 E_π は式 (2.19) のように表すことができる.

$$\begin{aligned} & \underset{\theta}{\text{maximize}} E_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} A_t \right] \\ & \text{subject to } D_{\text{KL}}^{\text{max}}(\theta_{\text{old}}, \theta) \leq \delta \end{aligned} \quad (2.19)$$

ここで, $\pi_{\theta_{\text{old}}}(a_t|s_t)$ はパラメータ更新前の方策, $\pi_\theta(a_t|s_t)$ は更新後の方策を表している. また, D_{KL} は KL ダイバージェンスのことであり, 分布間の距離を測る指標である. 式 (2.19) における $D_{\text{KL}}^{\text{max}}(\theta_{\text{old}}, \theta)$ は, 任意のパラメータの組み合わせに対して, KL ダイバージェンスを算出した時の最大値を表している. これは, 実際に用いる際にはパラメータの組み合わせ数が膨大になり, 最大値の算出が困難になる. そのため, 式 (2.20) に示すように, 平均値で代用する.

$$\begin{aligned} & \underset{\theta}{\text{maximize}} E_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} A_t \right] \\ & \text{subject to } \bar{D}_{\text{KL}}(\theta_{\text{old}}, \theta) \leq \delta \end{aligned} \quad (2.20)$$

ここで, $\bar{D}_{\text{KL}}(\theta_{\text{old}}, \theta) = E_{s \sim \rho}[D_{\text{KL}}[\pi_{\theta_{\text{old}}}(\cdot|s), \pi_\theta(\cdot|s)]]$ である. つまり, TRPO では更新前の方策 $\pi_{\theta_{\text{old}}}(\cdot|s_t)$ と, 更新後の方策 $\pi_\theta(\cdot|s_t)$ との距離を δ 以下となるように制約を施している. また, TRPO における KL ダイバージェンスを用いた制約は, 式 (2.21) に示すように, 目的関数に組み込むことができる.

$$\underset{\theta}{\text{maximize}} E_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} A_t - \beta \bar{D}_{\text{KL}}(\theta_{\text{old}}, \theta) \right] \quad (2.21)$$

■ Proximal Policy Optimization [41]

TRPO の式 (2.21) では、 $\bar{D}_{\text{KL}}(\theta_{\text{old}}, \theta)$ を減算している。そのため、 $\bar{D}_{\text{KL}}(\theta_{\text{old}}, \theta)$ の値が大きい場合、Advantage A_t の最大化が困難になってしまう。この問題を解決するために、Proximal Policy Optimization (PPO) が提案されている。PPO は、方策の目標値をクリッピングすることで、方策の更新を制約する手法である。TRPO と同様に、方策勾配法の学習結果が安定しないという問題に対し改善を図った手法である。TRPO では、KL ダイバージェンスを制約として用いていた。それに対し、PPO では式 (2.22) に示すように、目的関数を $L^{\text{CLIP}}(\theta)$ として勾配を求める。

$$L^{\text{CLIP}}(\theta) = E_t[\min(r_t(\theta)A_t, \text{clip}(r(\theta), 1 - \epsilon, 1 + \epsilon)A_t)] \quad (2.22)$$

ここで、 $\text{clip}(r(\theta), 1 - \epsilon, 1 + \epsilon)$ は、 $r(\theta)$ を $1 - \epsilon$ から $1 + \epsilon$ の範囲に制限する関数である。また、 $r_t(\theta)$ は確率 (方策) の比率を表し、式 (2.23) のように表される。

$$r(\theta) = \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}, \text{ so } r(\theta_{\text{old}}) = 1 \quad (2.23)$$

このように PPO では、方策の目標値をクリッピングすることで、TRPO の問題を回避しながら学習の安定化を実現している。

2.1.4 Actor-Critic ベースにおける代表的な深層強化学習手法

Actor-Critic ベースな深層強化学習とは、方策の学習方法を Actor-Critic 法ベースとした深層強化学習アルゴリズムである。本節では、Actor-Critic ベースの代表的な深層強化学習手法として、Asynchronous Advantage Actor-Critic (A3C) [44] について述べる。

■ Asynchronous Advantage Actor-Critic [44]

Asynchronous Advantage Actor-Critic (A3C) とは、学習で用いる経験の生成を並列実行することで高速化し、パラメータの更新を非同期的に行う手法である。マルチコア CPU の単一マシン上で学習が可能であり、Atari2600 において短時間で高いスコアを達成した。

図 2.7 に A3C の構造を示す。ここで、経験の生成を並列実行する機構を worker と呼ぶ。A3C のネットワークには、各 worker 独自に持つ local network と全 worker が共有している global network が存在する。これらのネットワークは、Actor と Critic から構成され、global network では Actor のパラメータ θ 、Critic のパラメータ θ_v を持ち、local network では Actor のパラメータ θ' 、Critic のパラメータ θ'_v を持つ。学習時に Actor の勾配 $d\theta$ と Critic の勾配 $d\theta_v$ を worker 毎に蓄積し、その勾配の積算値 $d\theta$ 、 $d\theta_v$ を非同期的に global network へ送ることで学習する。

A3C の損失関数は状態価値と方策の損失関数で表すことができる。状態価値の損失関数 L_v を式 (2.24)、方策の損失関数 L_p を式 (2.25) に示す。式 (2.24) で、 θ^- は更新前のネットワークのパラメー

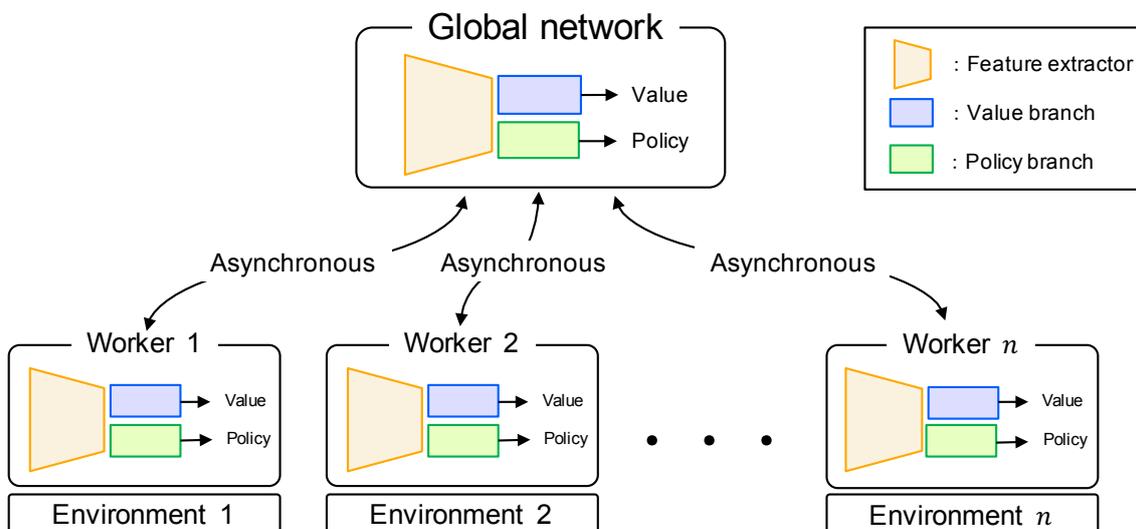


図 2.7: A3C の構造

タである。式 (2.25) で、 β はエントロピーの正則化項の強さを制御するパラメータである。 $H(\pi)$ は局所的な最適解に収束しないように、探索を促進するためのエントロピーである。

$$L_v = (r + \gamma V(s_{t+1}, \theta^-) - V(s_t, \theta))^2 \quad (2.24)$$

$$L_p = -\log(\pi(a|s))A(s, a) - \beta H(\pi) \quad (2.25)$$

式 (2.24) で示す価値関数の損失関数と式 (2.25) で示す方策の損失関数の和によって、A3C の損失関数 L_{A3C} は表される。式 (2.26) に A3C の損失関数 L_{A3C} を示す。

$$L_{A3C} = L_v + L_p \quad (2.26)$$

Asynchronous 非同期的でマルチエージェントな分散学習のことである。CPU のマルチスレッドを用いて複数の学習環境を用意し、各環境でエージェントが学習を行い経験を積み重ねる。パラメータはスレッド毎で個別に持つパラメータ θ_{local} と、global network と共有するパラメータ θ_{global} を持つ。アルゴリズムとしては、第一に θ_{local} に θ_{global} を同期する。第二に θ_{local} を使い、 θ_{local} の更新量 $d\theta_{\text{local}}$ を計算する。第三に更新量 $d\theta_{\text{local}}$ で θ_{global} を更新する。これらの操作を繰り返し実行する。また、パラメータだけでなく、RMSprop の勾配の 2 乗の移動平均も global network と共有する。

Asynchronous のメリットとして、LSTM が使用できる点が挙げられる。過去の経験からランダムサンプリングした経験を用いてパラメータを更新する Experience replay を使わず、各スレッドが現在の経験から学習する。そのため、時系列情報を考慮することができる。

Advantage A3C の Advantage には、2 つの意味がある。1 つ目は「価値の更新を 1 ステップ先ではなく 2 ステップ以上先まで考慮すること」と、2 つ目は「行動の相対的な価値を用いる」という意味である。まず、はじめに 1 つ目の意味である「価値の更新を 1 ステップ先ではなく 2 ステップ以

上先まで考慮すること」について説明する。式 (2.27) に 1 ステップ先の行動価値の更新式、式 (2.28) に 2 ステップ先を考慮した行動価値の更新式を示す。式 (2.28) で、 s は行動前の状態、 s' は行動後の次状態を表す。また、 $Q(s, a)$ は状態 s のとき行動 a を選択した場合の行動価値、 r は報酬、 γ は行動価値にノイズや遅れがあることを考慮し割り引くもので割引率と呼ぶ。

$$Q(s, a) \leftarrow r(t) + \gamma \max[Q(s', a)] \quad (2.27)$$

$$Q(s, a) \leftarrow r(t) + \gamma r(t+1) + \gamma^2 \max[Q(s', a)] \quad (2.28)$$

しかし、Advantage では行動 a_{t+1} を決定する際、計算途中の行動価値を使用することになる。したがって、その行動価値が間違っていた場合、その後の行動価値も間違えることになる。そのため、どの程度先まで考慮するかが重要になる。また、A3C では Actor-Critic 法が用いられているため、2 ステップ以上先まで考慮する価値は行動価値 $Q(s, a)$ ではなく、状態価値 $V(s)$ である。

次に、2 つ目の意味である「行動の相対的な価値を用いる」ことについて説明する。これは、行動価値から状態価値を差し引いて行動を評価する方法である。状態における行動価値は、行動よりも状態の影響を強く受ける。例えば、通勤時に電車が遅延していた場合、どのような行動をとっても遅れてしまう。このように価値は状態に依存している傾向にある。ここで、行動価値 $Q(s, a)$ から状態価値 $V(s)$ を差し引くことで、行動の相対価値を算出する。行動の相対価値 $A(s, a)$ を式 (2.29) に示す。

$$A(s, a) = Q(s, a) - V(s) \quad (2.29)$$

この $A(s, a)$ を Advantage 関数と呼ぶ。そのため、Advantage 関数を用いた場合の方策の勾配 $\nabla J(\theta)$ は、式 (2.30) のように表せる。

$$\nabla J(\theta) = E[\nabla_{\theta} \log \pi_{\theta}(a|s) A(s, a)] \quad (2.30)$$

2.1.5 Transformer 構造を導入した深層強化学習手法

2017 年に自然言語処理分野で Transformer [2] が提案されて以降、Transformer アーキテクチャは教師あり学習分野の様々なタスクで目覚ましい性能を発揮している [51, 52, 53, 54, 55]。Transformer は、従来の Convolutional Neural Network (CNN) や Recurrent Neural Network (RNN) よりも、時系列データの長期的な依存関係を捉える能力に優れている。そのため、時系列特徴を考慮した動的な制御が求められる深層強化学習分野でも有効性が期待されており、Transformer 構造を採用した深層強化学習手法も数多く提案されている [3, 56, 57, 58, 59, 60, 61, 62, 63]。本節では、まず Transformer について述べたのちに、深層強化学習の文脈で Transformer 構造を導入した先駆けである Decision Transformer について述べる。

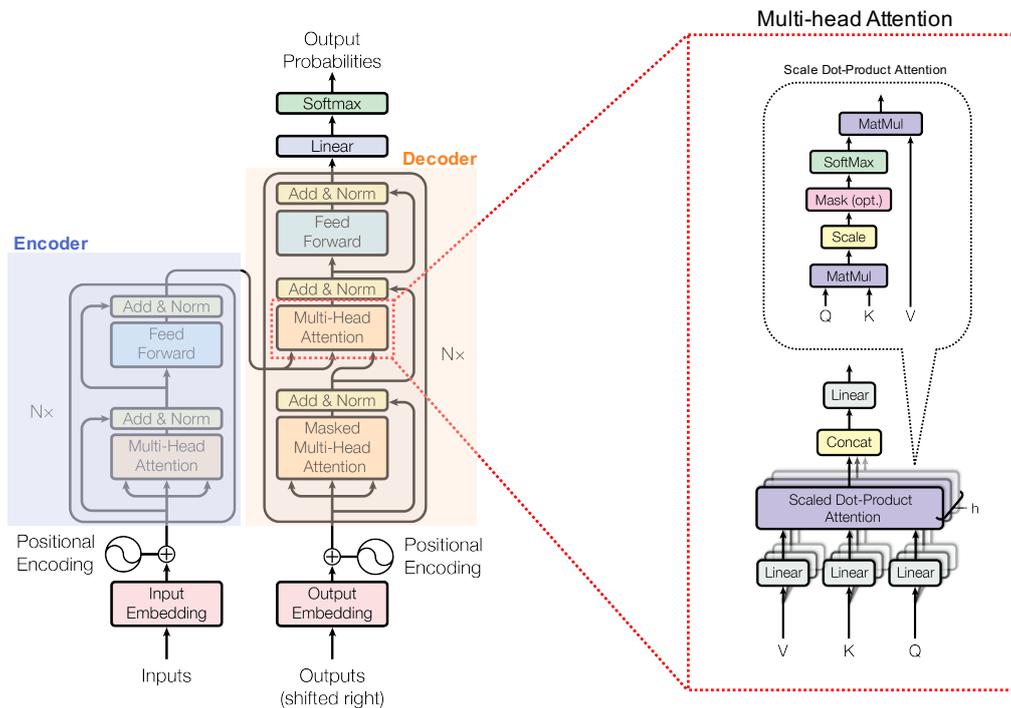


図 2.8: Transformer アーキテクチャ (文献 [2] から抜粋)

■ Transformer [2]

Transformer は、自然言語処理において提案されたモデルであり、Attention だけで入力文と出力文の文章同士に対する広範囲な依存関係を捉えられるアーキテクチャである。その特徴として、RNN のような再帰構造や CNN などの畳み込み構造を一切使用しない点、演算の並列化が可能であり学習時間を短縮した点、他タスクにも適用可能な高い汎用性を持つ点などが挙げられる。これらの特徴から、近年の大規模言語モデル (Large Language Model; LLM) の開発を促進し、Generative Pre-trained Transformer (GPT) [17] をはじめ、多くの LLM はこの Transformer アーキテクチャを基盤に構築されている。

Transformer アーキテクチャ Transformer は、入力情報から特徴を抽出する Encoder と、出力値を算出する Decoder で構成されている。図 2.8 に Transformer アーキテクチャを示す。以下では、Transformer の代表的なモジュールについて述べる。Multi-Head Attention とは、入力間の関係性を考慮する Scaled Dot-Product Attention を複数並列に構築したモジュールである。このモジュールでは、まず 3 つの線形層 (Linear) により入力行列を Query, Key, Value のベクトルに変換する。入力行列 X に対する Query, Key, Value ベクトル Q, K, V への行列変換は次のように表現される。

$$Q = XW_Q, K = XW_K, V = XW_V \quad (2.31)$$

ここで、 W_Q, W_K, W_V はそれぞれの線形層の重みパラメータである。これらベクトルを複数の Scaled Dot-Product Attention に適用し、それら計算結果を統合することで出力を得る。Scaled Dot-Product

Attention では、Query と Key から内積にもとづいた attention weight を算出し、この重みと Value 間の要素積を取る行列演算を行う。この演算はあくまで内積による attention の計算であるため、学習パラメータは持たない。そのため、Scaled Dot-Product Attention 前で Linear によって入力値を Query, Key, Value ベクトルに埋め込むことで学習可能にしている。Scaled Dot-Product Attention における行列演算は、式 (2.31) のように表現できる。

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (2.32)$$

Q, K, V は d_k 次元の Query, Key, Value ベクトルであり、 K^T は Key ベクトルの転置行列を意味する。softmax(\cdot) は、softmax 関数である。ここで、softmax 関数はベクトルに対する行方向、つまり各 Query に対して全ての Key への重みの総和が 1 になるように適用される。このように算出された attention weight は、Query と Key 間の類似度を示す役割を担い、Query から Key へ注視するパターンを表現している。この Scaled Dot-Product Attention を複数に並列で構築、つまりマルチヘッドにすることで、モデルの表現力向上を図っている。

上述の Scaled Dot-Product Attention は、Query と Key ともに同一の入力行列 X に対する演算である。このような同じ入力行列の要素間に対する attention weight の演算を「Self-attention」と呼ぶ。一方で、異なる入力行列 X, Y の要素間に対する attention weight の演算を「Cross-attention」と呼ぶ。Encoder の Multi-Head Attention は Self-attention, decoder の Multi-Head Attention は Cross-attention を意味する (図 2.8 参照)。また、文章生成のような自己回帰タスクの場合、self-attention の計算時に各要素 (単語) が自身より未来の要素 (単語) を参照してしまうため、未来の要素への参照を制限する必要がある。この各要素が未来の要素へアクセスできないようにマスク処理を施した Multi-Head Attention を Masked Multi-Head Attention と呼ぶ。

Multi-Head Attention 以外には、Multi-Head Attention の出力をトークンごとに変換する順伝播型ニューラルネットワークである point-wise feed forward network (Feed Forward) や、加算と正規化を施す Add & Norm, 残差接続 (residual connection) といったモデルの学習効率を向上させる構造を採用している。また、RNN や CNN では学習によって特徴ベクトルの位置情報を学習可能である [64, 65]。しかし、transformer はモデル構造上の問題により、入力ベクトルの位置情報をモデルそのものが学習によって考慮できない。そのため、入出力ベクトルの位置情報を表現したベクトルを入力ベクトルに付加する Positional Encoding が導入されている。この位置ベクトルは sin/cos 波を採用することが多く、入力ベクトルが文章データの場合は文中の何番目の単語か、画像データの場合は画像内のどのピクセルか、時系列データの場合は何時刻目かを表現している。

Transformer の性質 Transformer モデルは、CNN のように局所的な特徴抽出ではなく、大域的な特徴抽出が可能であることから、モデル構造として帰納バイアスが低く高い汎用性を保有する。一方で、学習データ量が不十分だと過学習が起きやすいデメリットを持つ。また Transformer モデルの性能は、モデルのパラメータ数、学習データ数、学習計算量の 3 要素によって記述される冪乗則にしたがって、線形的に性能が向上する性質を持つことが報告されている [66]。この性質は画像やマルチモーダル、強化学習を含めた他ドメインでも報告されている [67, 68]。

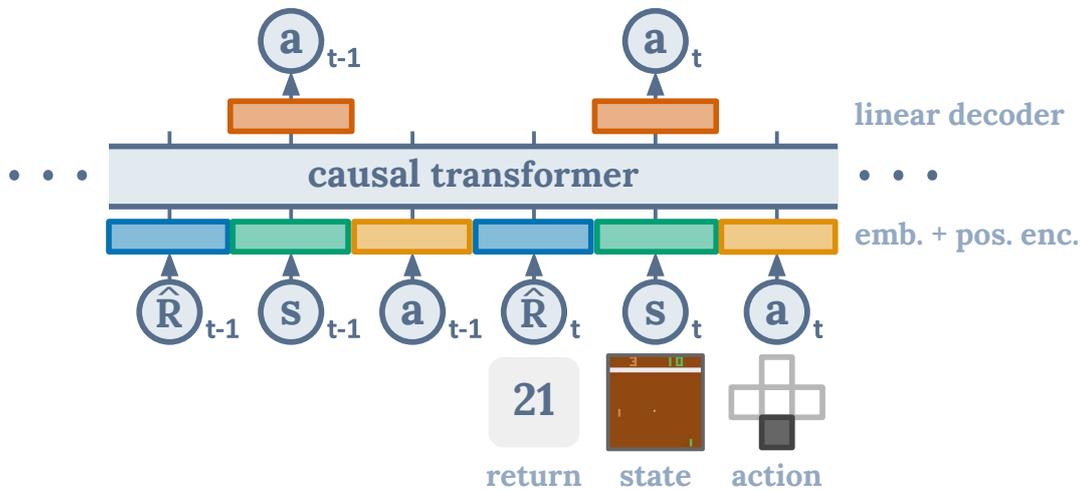


図 2.9: Decision Transformer の概要 (文献 [3] から引用)

■ Decision Transformer [3]

Decision Transformer は、GPT アーキテクチャを採用したオフライン深層強化学習手法で、深層強化学習分野で Transformer 構造を導入した先駆けの手法である。図 2.9 に decision transformer の概要を示す。GPT などの言語モデルは、入力された文章に続く文章/単語が何であるかを逐次的に予測している。この点に着目し、decision transformer は強化学習のエージェントと環境のインタラクション (状態観測, 行動選択, 報酬獲得, 次状態観測) において、ある状態に続く行動は何であるかを逐次的に予測するタスクと定義している。これにより、言語モデルによる文章生成と同じ枠組みで、エージェントの行動を生成するモデルを提案している。Decision transformer ではエージェントの行動を逐次的に予測するため、モデルの入力値は報酬, 状態, 行動であり、出力値は行動である。ここで、報酬はスカラー, 状態や行動は画像や数値情報などのベクトルであり、データ形式が様々であるため、報酬, 状態, 行動それぞれに対する異なる埋め込み層を構築し、transformer への入力次元を揃えることで対処している。強化学習が対象とする問題では、エージェントの振る舞いによる影響が即時的に得られるとは限らず、この行動と報酬間の関係性に対する学習に時間を要していた。この問題に対し、transformer 構造による逐次的な行動予測は、transformer の大域的特徴抽出によって、離れた時刻同士の行動と報酬の関係性を効率的に捉えることができ、エージェントの高い制御性能を獲得している。

この decision transformer の登場によって、深層強化学習分野でも transformer 構造を利用した手法が数多く提案されている。例えば、transformer の汎用性に着目し、atari2600 の様々なビデオゲームで学習させた Multi-game Decision Transformer [58] や、ビデオゲームに限らず多種多様な意思決定タスクを用いた教師あり学習による汎用エージェントである Gato [59], メタ学習へ発展させた Algorithm Distillation [69] が挙げられる。

2.2 深層強化学習の応用事例

深層強化学習は、多くの研究者や開発者によって大きな技術進歩を遂げ、様々なタスクで高い制御性能を示した。そのため、実世界の意思決定が求められるシステムを軸とし様々な分野に適用されている。以下では、2.1 節で挙げたような深層強化学習手法が、どのような実問題に適用されているか応用事例について述べる。

ゲーム攻略。 ゲーム攻略は、深層強化学習の応用先として最もよく知られたタスクである。Atari2600 のビデオゲームは、深層強化学習アルゴリズムの制御性能のベンチマーク Atari 100k benchmark [70] として普及し、様々な深層強化学習アルゴリズムの評価に用いられている。囲碁 AI である AlphaGo/Zero [6, 7] も深層強化学習が適用された例である。AlphaGo/Zero は、囲碁における人間のプレイデータにもとづく教師あり学習の後、強化学習を用いた自己対戦を行うことで、エキスパートを凌駕する性能を獲得している。また、レーシング AI エージェントである Gran Turismo Sophy は、グランツーリスモというレーシングゲームにおいて深層強化学習による運転技術を学習し、世界のトップドライバーと対等に競争できる非常に高レベルな運転技術を獲得している [71]。

ロボティクス。 ロボティクスもゲーム攻略と同様に高い制御性能が求められるため、深層強化学習の応用例として広く知られている。深層強化学習エージェントによる単腕アームロボットのロボットマニピュレーションタスク [8] や、障害物を回避しながら自律的に目的地に到達する移動ロボットのナビゲーションタスク [72]、ドローン制御など多くのロボット制御タスクに深層強化学習が応用されている。また、ロボット制御タスクで深層強化学習を実行するライブラリとして、Mujoco [73] や Issac-gym [74] が提供され、ロボティクス分野における深層強化学習の応用に期待されている。

自動運転。 自動運転には認知、判断、制御の3つがあり、シーンに合わせた車両制御が求められる。そこで、深層強化学習による高い制御性能が期待されている。深層強化学習モデルによる end-to-end での車両制御システム [9] が開発されている一方で、シーンごとに特化させた深層強化学習による車両制御モデルも開発されている。例えば、駐車を目的とした深層強化学習モデル [75] や、高速道路での合流を目的としたモデル [76]、交差点での制御を目的としたモデル [77] などが挙げられる。

ヘルスケア。 医療分野では、患者の状態を監視し、患者の個人に適した医療の提供する動的治療計画 (Dynamic Treatment Regime; DTR) と呼ばれる治療法がある。この治療法は強化学習の枠組みに似ており効果が見込まれるため、深層強化学習により最適化されたシステムによって患者個人に適した医療を提案する仕組みが開発されている [78]。この DTR に対する深層強化学習の応用は、治療の即時的な効果だけでなく長期的な効果も考慮して、患者に適した治療の提案が可能となる。また、糖尿病やがんなどの慢性疾患は、患者の健康に対する不適切な習慣が原因で発症および重症化していると考えられている。この問題に対し、患者に対して慢性疾患の予防や治療を促す行動を促進させることを目的とし、患者の生活に対し適切に介入する深層強化学習モデルが開発されている [79]。この手法においても患者個人の特性を鑑みた介介入行動の獲得を実現している。

自然言語処理。 自然言語処理分野では、大規模言語モデルのアライメントに深層強化学習が応用されている。この言語モデルのアライメントを Reinforcement Learning from Human Feedback (RLHF)

[10] と呼ぶ。これは言語モデルが生成する文章の質を向上させるために、人間のフィードバックを用いて言語モデルの出力を強化学習により調整する手法である。大規模言語モデルには、虚偽情報を真実であるかのように文章生成してしまうハルシネーションや、ユーザにとって好ましくない発言（設計者が意図しない発言）をしてしまうなどの課題がある。これらの対処として、RLHF を大規模言語モデルの Post-Training として行なっている。ChatGPT はこの RLHF を多段階に実行し、ユーザにとって好ましい高性能な質問応答性能を獲得している。

その他。 金融分野では、特定の目的を最大化するために最適な資産分配を行うポートフォリオマネジメント [80] や、利益を得ることを目的として証券や金融商品を売買するトレーディング [81] などに深層強化学習が応用されている。核融合では、核融合反応が発生している間、強力な磁場を利用してプラズマを空間内に閉じ込め維持する必要がある、この制御コイルの自律制御に応用されている [82]。航空分野では、航空機管制による飛行計画やスケジュール調整 [83]、ロケットエンジンの制御 [84]、軍用機の本ミサイル制御 [85] などで応用されている。他には、頻繁に変化するユーザの好みに合わせたニュースの推薦システム [86] や、データセンターにおける冷却システムの制御 [87] などが深層強化学習の応用例として挙げられる。

2.3 深層強化学習の課題

深層強化学習の代表的な課題として、「探索と利用のトレードオフ」、「報酬設計の困難さ」、「長期戦略を獲得が困難」、「環境の部分的観測への対応」、「学習効率の低さ」、「エージェントの信頼性」が挙げられる。以下でこれらの課題に関して詳細を述べる。

探索と利用のトレードオフ。 深層強化学習では、環境に対する探索行動（探索）を試行することで高い報酬が得られる可能性があるが、探索ばかりでは一向に方策が最適化できない。一方で、既知の行動（利用）に固執しすぎてもより良い状態に到達できず、方策の最適化が妨げられてしまう。これら探索と利用のバランスを適切に取るのが非常に重要な要素の一つである。この課題に対する手法として、 ϵ -greedy 法を始めとし、エージェントの内発的動機付け [88, 89, 90, 91] などが挙げられる。

報酬設計の困難さ。 深層強化学習エージェントの性能は、報酬の設計に大きく依存し、報酬設計が適切でないとエージェントが望ましくない方策を獲得してしまうことがある。そのため、目標に適した網羅的な報酬設計が必要であるが、これら適切な報酬関数を見つけることは非常に困難である。この課題に対する手法として、逆強化学習 (Inverse Reinforcement Learning; IRL) [92, 93, 94, 95, 96, 97] などが挙げられる。

長期戦略の獲得が困難。 深層強化学習では、エージェントが一連の行動を通して受け取った最終的な報酬に対して、どの行動がどれだけ貢献したのかを特定することが困難である。特に長期的なタスクでは、ある行動が将来的な結果にどのように影響を与えたかを見極めることが非常に困難である。この課題に対する手法として、階層型強化学習 (Hierarchical Reinforcement Learning; HRL) などが提案されている [98, 99, 100]。

環境の部分的観測への対応. 多くの実問題では、エージェントが環境の全ての情報を直接観測できない場合が多く発生する。エージェントが一部の情報しか観測できないことで、状態に対する評価の不確実性が増し、最適な方策を学習することが困難となる。本課題は、深層強化学習エージェントの環境に対する観測が、部分観測マルコフ決定過程 (Partially Observable MDP; POMDP) にしたかった問題設定である。この課題に対する手法として、不確実性を考慮したエージェントの内部状態表現 [101, 102, 103, 104] などが挙げられる。

学習効率の低さ. 深層強化学習における学習は、環境との大規模な試行錯誤を通じてデータを収集し学習するが、これには計算資源と時間が必要となる。深層強化学習では、この学習効率を改善することが大きな課題の一つとなっている。この課題に対して、model ベースな手法である世界モデルを用いた手法 [105, 70, 106, 107] や、メタ強化学習 [108, 109, 110, 111]、学習タスクの追加である教師なし学習やマルチタスク学習によるメインタスクに対する学習効率向上 [11, 12, 112, 13] などが提案されている。

エージェントの信頼性. 深層強化学習は 2.2 節で述べた通り、様々な制御問題で高い性能を示している。一方で、深層学習が持つモデルのブラックボックス性や、報酬設計や探索効率に起因するエージェントの予期せぬ動作の学習など、シミュレーション上ではなく実問題に適用する際には信頼性に関する多くの課題が残っている。この問題については、2.4 節で詳細に述べる。

本節では、「学習効率の低さ」に対するアプローチの一種であるマルチタスク学習に焦点を当て、いくつかの研究を紹介する。深層強化学習の文脈では、解きたい目標 (メインタスク) があり、その目標を達成するための学習データ収集に対するサンプル効率が課題となっている。この課題の対処として、マルチタスク学習の中でも補助学習によるメインタスクの学習効率向上に関する研究が報告されている [113, 114, 115, 116, 13]。この補助学習とは、メインタスクとは異なった目標のタスク (補助タスク) を、メインタスクと同一モデルで学習することで、メインタスクに対する学習を促進させる方法である。Mirowski らは、深層強化学習による 3D 環境でのナビゲーションをメインタスクとし、RGB 画像から奥行き情報を予測する depth prediction という補助タスクを導入することでナビゲーション性能の向上を報告している [115]。また Kartal らは、深層強化学習の枠組みにおけるエピソードの終端状態に焦点を当て、現状態が終端状態にどれだけ近いかを予測する Terminal Prediction (TP) という補助タスクを提案している [114]。Hernandez-Leal らは、複数エージェントの制御を目的としたマルチエージェント問題において、他エージェントの行動を予測する補助タスクである agent-modeling を提案している [116]。このように多くの研究で、深層強化学習の枠組みに適した補助タスクが提案されている。以下で、補助タスクを用いた深層強化学習エージェントの学習効率向上を目的とした代表的な手法である UNREAL [13] について述べる。

■ Unsupervised REinforcement learning and Auxiliary Learning [13]

Unsupervised REinforcement learning and Auxiliary Learning (UNREAL) は、教師なし学習の補助タスクを同時に行う強化学習の手法である。図 2.10 に UNREAL のネットワーク構造を示す。UNREAL

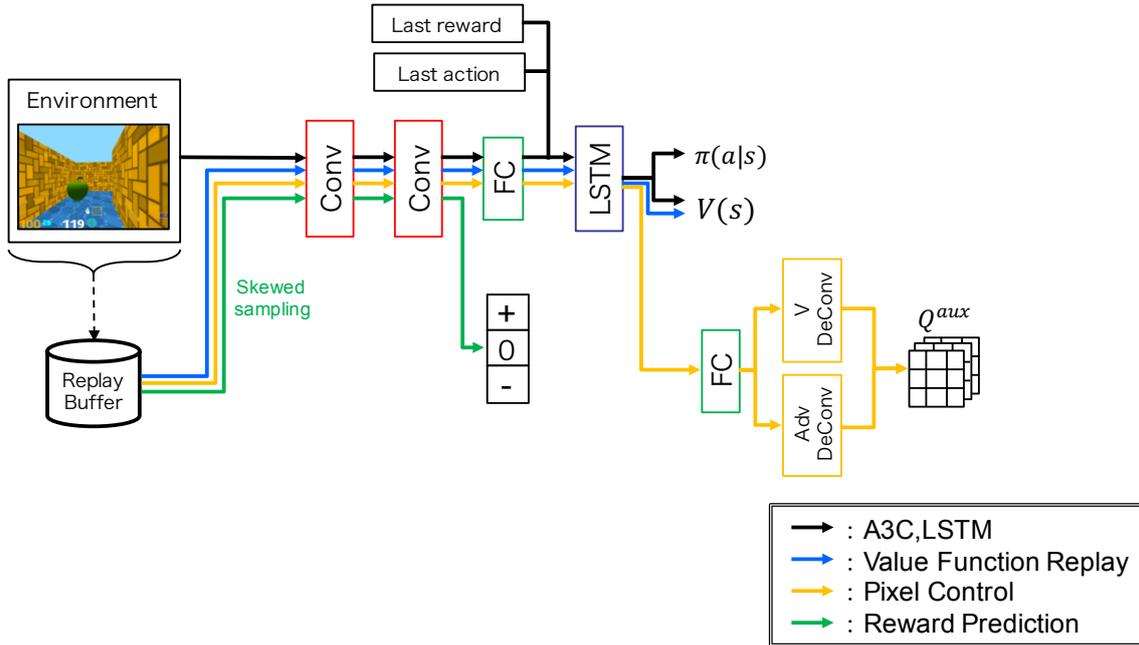


図 2.10: UNREAL のネットワーク構造

のネットワークは、ベースの A3C と補助タスクの Pixel Control, Reward Prediction, Value Function Replay から構成されている。また、各補助タスクのネットワークは、A3C のネットワークと一部共有することで、A3C によって解くタスクの補助する役割を担っている。UNREAL における損失関数は、式 (2.26) で表される A3C の損失関数 L_{A3C} と、各補助タスクの損失関数との和で表される。UNREAL の損失関数 L_{UNREAL} を式 (2.33) に示す。

$$L_{UNREAL}(\theta) = L_{A3C} + \lambda_{VR} L_{VR} + \lambda_{PC} \sum_c L_Q^{(c)} + \lambda_{RP} L_{RP} \quad (2.33)$$

ここで、 L_{VR} は Value Function Replay の損失関数、 $\sum_c L_Q^{(c)}$ は Pixel Control の損失関数、 L_{RP} は Reward Prediction の損失関数である。また、 $\lambda_{VR}, \lambda_{PC}, \lambda_{RP}$ は各補助タスクの重み付け項である。以下でこれら 3 つの補助タスクについて説明する。

Pixel Control 画像の画素が大きく変化する行動を学習する補助タスクである。入力画像をグリッド毎に分割し、各グリッド毎で画素値が大きく変化する行動を n-step Q 学習により学習する。Pixel Control のネットワークは、畳み込み層 2 層、全結合層、LSTM、全結合層で構成されている。行動価値関数 Q の推定に Dueling Network を組み込んでいる。また、入力画像に畳み込みを行い Deconvolution することで入力画像を荒くし、画像の特徴量を抽出しやすくしている。

Pixel Control は学習に n-step Q 学習を用いるため、損失関数は n-step Q 学習の損失関数である。Pixel Control の損失関数 $L_Q^{(c)}$ を式 (2.34) に示す。式 (2.34) で、 c は入力画像を分割したグリッド数、 θ^- は一つ前のパラメータ、 γ は割引率を表す。

$$L_Q^{(c)} = (r + \gamma \max_{a'} Q^{(c)}(s', a' : \theta^-) - Q^{(c)}(s, a, : \theta))^2 \quad (2.34)$$

Value function Replay 状態価値関数 $V(s)$ の学習に過去の経験をシャッフルし学習させる補助タスクである。この補助タスクは、従来手法である Deep Q-Network (DQN) で用いられる Experience replay と同様の役割を果たす。Value Function Replay のネットワークは、畳み込み層 2 層、全結合層、LSTM で構成されている。

Value Function Replay の損失関数は A3C と同様に 2 乗誤差を用いている。Value Function Replay の損失関数 L_{VR} を式 (2.35) に示す。

$$L_{VR} = (r + \gamma V(s_{t+1}, \theta^-) - V(s_t, \theta))^2 \quad (2.35)$$

Reward Prediction 報酬が獲得できた状態を多く学習させ、現在の状態から未来の報酬を予測する補助タスクである。実際に報酬が獲得できた状態を Replay Buffer からサンプリングし入力とする。Reward Prediction のネットワークは畳み込み層 2 層で構成されている。

Reward Prediction の損失関数は交差エントロピー誤差が用いられている。Reward Prediction の損失関数 L_{RP} を式 (2.36) に示す。式 (2.36) で、 t_{target} は実際に得られた報酬値 (教師データ)、 y は Reward Prediction により予測した報酬値である。

$$L_{RP} = - \sum t_{target} \log y \quad (2.36)$$

2.4 説明可能な強化学習に関する研究動向

深層強化学習エージェントは高い制御性能を獲得できることが知られており、様々なタスクへの応用が期待されている。一方で、この高性能化の背景には、深層学習の恩恵が大きく関わっており、説明可能な AI (eXplainable AI; XAI) 分野と同様に、エージェントモデル内部の処理に用いるパラメータ数の増加と複雑化に伴い、エージェントモデルのブラックボックス化が問題となっている。このブラックボックス化は、エージェントモデルがなぜそのような行動を選択したか、判断理由が不明となってしまう問題であり、深層強化学習の信頼性に直結する重大な課題である。このような強化学習における説明性や解釈性向上を目的とした研究分野を XAI になぞらえて「説明可能な強化学習 (eXplainable Reinforcement Learning; XRL)」と呼ばれている。

本節では、この XRL に関する研究を体系的に述べる。XRL の研究は、主に「interpretable agent (IA)」、「intrinsic explainability (IE)」、「post-hoc explainability (PHE)」にカテゴライズすることができる。表 2.1 に XRL に関する研究のカテゴライズを示す。interpretable agent (IA) とは、人間が理解可能である単純な関数や数式にエージェントモデルを近似することや、エージェントモデルの行動選択に人間が設定したルールを課すことで説明を獲得するアプローチのことである。例えば、決定木を用いたエージェントモデルは、この決定木そのものがエージェントモデルに対する説明を提供している。これらのアプローチは、状態空間の狭いような単純な環境下では上手く機能し、人間にとって理解しやすい説明の提供が可能である。しかし、複雑な環境下では決定木が深くなりすぎることや、ルールの複雑化など、人間が理解しやすい説明の提供が困難になる傾向がある。intrinsic explainability (IE)

表 2.1: 説明可能な強化学習に関する研究のカテゴリライズ

categories	methods
interpretable agent	[117], [118], [119], [120], [121]
intrinsic explainability	[122], [123], [124], [5], [125], [126], [127], [128]
post-hoc explainability	[129], [130], [131], [4], [15], [132], [133], [134], [135], [136], [137]

とは、エージェントや報酬関数などのモデル構造を説明性を提供するモジュールを追加・変更することで、強化学習の枠組みを説明可能とするアプローチである。例えば、エージェントモデルへの attention module の導入し、エージェントの注視領域を示す顕著性マップを生成する。これらのアプローチは、エージェントモデルの行動選択や報酬モデルの報酬算出に直接作用するため、意思決定プロセスに対する忠実な説明を提供することができる。また、これらの説明はモデルの順伝播時に生成されるため、計算コストを低く抑えることができる。post-hoc explainability (PHE) とは、解釈不可能なエージェントモデルや報酬モデルなどに対し、モデル構造を変更せず特定の後処理にて説明を生成するアプローチである。例えば、エージェントモデルの入力値に摂動を付加し、モデルの出力値に対する揺らぎからエージェントモデルの注視領域を特定する。これらのアプローチは、エージェントや報酬関数などのモデル構造に対する変更を必要としないため、高い性能を有しているブラックボックスモデルに対する説明の生成に有効である。一方で、このような後処理にて説明を提供するアプローチは、摂動や勾配などにもとづき説明を生成するため、計算コストは高い傾向がある。

深層強化学習におけるエージェントモデルは、ニューラルネットワークによって構成されることが一般的であり、モデルの意思決定が複雑である。そのため本節では、IE と PHE に焦点を当て、それぞれの視覚的説明と言語説明について詳細に述べる。

2.4.1 エージェントモデルに対する視覚的説明

エージェントモデルに対する代表的な視覚的説明手法として、摂動を用いるアプローチと、attention を用いるアプローチがある。

摂動を用いるアプローチ. Greydanus らは、エージェントモデルの入力に対して、ガウシアンばかしにもとづく摂動を付加し、エージェントモデルの出力に対する揺らぎを計測することで、エージェントモデルを視覚的に解析している [4]。Iyer らは、エージェントモデルの入力に対し、テンプレートマッチング法を用いて背景領域を特定し、オブジェクトに対し摂動を付加することで、オブジェクトレベルの視覚的説明を生成している [131]。Pan らは、DNN を用いてエキスパートの行動データを模倣する Generative Adversarial Imitation Learning (GAIL) に着目し、エキスパートモデルに摂動ベースな説明手法を適用することで、ローカル説明とグローバル説明を生成する説明可能な GAIL (xGAIL) を提案している [134]。

attention を用いるアプローチ. Zambaldi らは、オブジェクト同士の関係性を考慮する attention ベースな relational module をエージェントモデルに導入することで、環境内のオブジェクトの関係

性を可視化した [124]. 入力空間内の様々なオブジェクト同士がそれぞれ何に注目しているかを示すマップから、エージェントモデルを視覚的に解析している. Bao らは、自動運転分野に着目し、運転制御エージェントモデルに CNN ベースな saliency model を導入する DRIVE 提案している. 推論時のボトムアップな attention と foveal vision によるトップダウンな attention から、運転モデルに対する視覚的な説明を生成している [126]. ここで foveal vision とは、視覚システムにおいて視野の中心部分を注視した視覚情報のことである. Zhang らは、エージェントモデルの意思決定プロセスをタスク関連領域の特定と行動選択の 2 つに分割し、attention module を導入することで、タスクに関連する特徴に対する視覚的な説明を生成している [127]. Shi らは、タスク関連領域を強調する attention mask を生成する self-supervised interpretable network (SSINet) を提案し、これをエージェントモデルに導入することで、attention mask を用いてエージェントモデルを視覚的に解析している [136]. Mott らは、エージェントモデルに対し Query ベースの attention module を導入することで、タスクに関連する重要な情報である 2 種類の attention (「what」と「where」) を生成している [5].

画像認識分野における XAI 手法をエージェントモデルに応用したアプローチも報告されている. 画像認識分野で提案された CNN モデルの判断根拠を可視化する XAI 手法である Class Activation Map (CAM) [138] や Grad-CAM [139], Layer-wise Relevance Propagation (LRP) [140] を深層強化学習に適用しようと試みている. Weitkamp らは、Grad-CAM にもとづくボトムアップな注視領域算出法をエージェントモデルに適用することで、エージェントモデルの注視領域を示すマップを生成している [132]. Huber らは、エージェントモデルに対して LRP を導入することで、エージェントモデルの行動選択に対する顕著性マップを生成している [141]. He らは、CAM と Shapley Additive exPlanations (SHAP) [130] を統合して、航空機を制御する方策に対する視覚的説明を実現している [135]. SHAP とは、ゲーム理論における Shapley value をエージェントモデルの説明に応用した手法で、各特徴量に特定の予測に対するコストを割り当てることで、エージェントモデルの説明を生成している.

また上述以外のアプローチとして、ツールを用いるアプローチも報告されている. Zahavy らは、次元削減アルゴリズムである t-SNE を用いたハンドクラフトな特徴量エンジニアリングによって、エージェントモデルの特徴空間を可視化し、エージェントモデルが認識している状態を解析している [129]. Wang らと Jaunet らは、深層強化学習用に設計された視覚的分析ツール DQNViz および DRLViz を開発/提供している. これらのツールは、エージェントモデルをデバッグや改善できるようにすることを目的とし、学習中の Q 値の変化や、エピソードごとの方策の変化など、さまざまな観点での解析を可能としているを提供している [133, 142]

摂動を用いるアプローチと attention を用いるアプローチから、Greydanus らの視覚的説明手法と Mott らの視覚的説明手法をピックアップし、それぞれについて以下で説明する.

■ Greydanus らの視覚的説明手法 [4]

Greydanus らは、エージェントの行動選択が解釈できる顕著性マップを生成する摂動ベースの顕著性算出方法を提案している. 深層強化学習アルゴリズムには、2.1.4 節で述べた A3C を採用している. Greydanus らの視覚的説明手法は、特定の画素を削除した摂動をエージェントモデルの入力画像

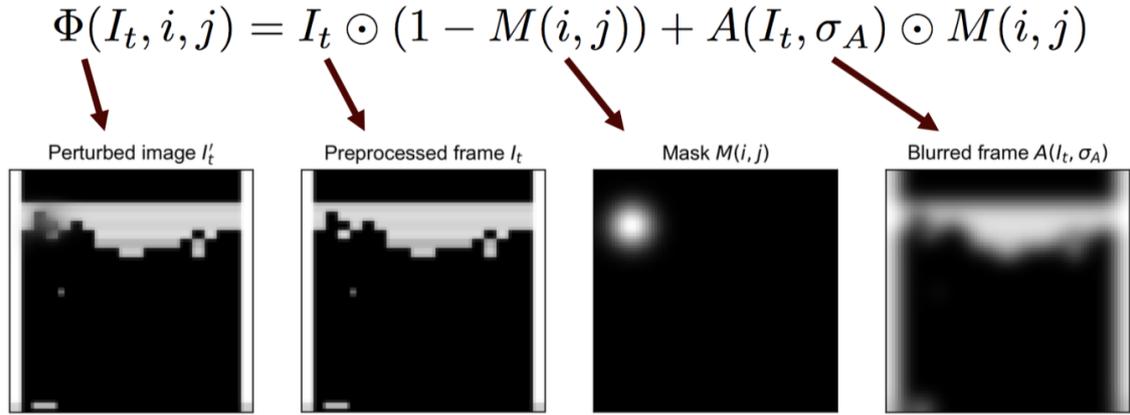


図 2.11: Greydanus らの視覚的説明手法における摂動画像例 (文献 [4] から引用, 式 (2.37) 参照)

に適用することで摂動画像を作成する. この摂動画像を用いて, 出力であるエージェントモデルの方策に対して影響する領域を確認することで, 削除した画素の重要度を算出する. 画像 I_t における画素 (i, j) を削除した摂動画像 $\Phi(I_t, i, j)$ を式 (2.37) に示す.

$$\Phi(I_t, i, j) = I_t \odot (1 - M(i, j)) + A(I_t, \sigma_A) \odot M(i, j) \quad (2.37)$$

ここで, $M(i, j)$ は画素 (i, j) を中心に分散 $\sigma^2 = 25$ の 2 次元ガウス関数 (ガウシアンぼかし) を施したマスク画像, $A(I_t, \sigma_A)$ は画像 I_t に対し標準偏差 σ_A のガウシアンぼかしを施した画像である. この摂動は, 画素 (i, j) 周辺の領域に不確かさを加えていると同義である. 式 (2.37) における各画像の例を図 2.11 に示す.

式 (2.37) により算出した摂動画像 Φ を用いて, 式 (2.38) に示すように, 入力画像の画素 (i, j) に対するスコア $S(i, j)$ を算出する.

$$S_\pi(t, i, j) = \frac{1}{2} \|\pi_u(I_{1:t}) - \pi_u(I'_{1:t})\|^2$$

$$\text{where } I'_{1:k} = \begin{cases} \Phi(I_k, i, j) & \text{if } k = t \\ I_k & \text{otherwise} \end{cases} \quad (2.38)$$

ここで, $\pi_u(I_{1:t})$ は softmax 関数を施す前の方策 π である. 式 (2.38) により, 画像 I_t 内の全画素に対してスコア $S_\pi(t, i, j)$ を算出することで, 時刻 t における方策 π のための顕著性マップを構築することができる. 式 (2.38) と同様に, 状態価値 V に対しても式 (2.39) のように定義することができる.

$$S_{V^\pi}(t, i, j) = \frac{1}{2} \|V^\pi(I_{1:t}) - V^\pi(I'_{1:t})\|^2 \quad (2.39)$$

式 (2.38), (2.39) により算出される方策と状態価値に対する顕著性マップ可視化することで, エージェントの行動選択を視覚的に解析可能としている.

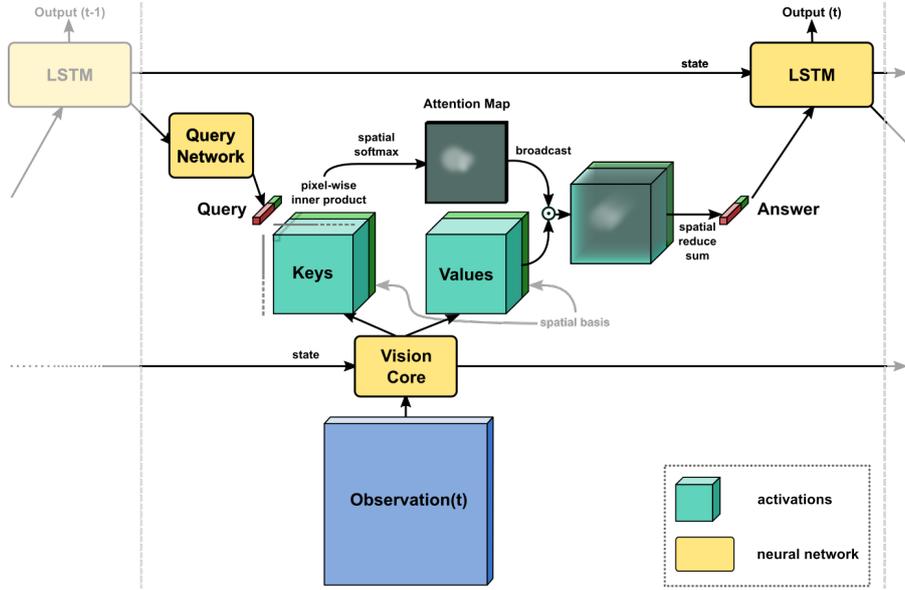


図 2.12: Mott らの視覚的説明手法におけるエージェントモデル構造 (文献 [5] から引用)

■ Mott らの視覚的説明手法 [5]

Mott らは、エージェントモデルに Query ベースの attention を導入することで、深層強化学習エージェントモデルの意思決定を視覚的に解析している。深層強化学習アルゴリズムとして、A3C の後続手法である Importance Weighted Actor-Learner Architecture (IMPALA) [47] を採用している。Mott らの視覚歴説明手法のエージェントモデル構造を図 2.12 に示す。Mott らの視覚歴説明手法は、Vision Core と Keys, Values, Query Network, LSTM から構成されている。式 (2.40) に示すように、時刻 t での環境からの観測値 $\mathbf{X}(t)$ (RGB 画像) を Vision Core へ入力し特徴抽出する。Vision Core は、複数の畳み込み層と Convolution LSTM (ConvLSTM) [143] などの RNN から構成され、特徴ベクトル \mathbf{O}_{vis} を生成する。

$$\mathbf{O}_{\text{vis}}, \mathbf{s}_{\text{vis}}(t) = \text{vis}_\theta(\mathbf{X}(t), \mathbf{s}_{\text{vis}}(t-1)) \quad (2.40)$$

Vision Core の特徴ベクトル \mathbf{O}_{vis} をチャンネル方向に 2 つのテンソルへ分割する。この 2 つのテンソルを、Keys \mathbf{K} と Values \mathbf{V} と呼ぶ。Query Network Q_ψ は、多層パーセプトロン (MLP) で構成されており、LSTM の隠れ状態 \mathbf{s}_{LSTM} を入力とし、クエリベクトル \mathbf{q}_n を生成する。クエリベクトルは、Keys \mathbf{K} のチャンネル数 N と一致するように生成される。Query Network Q_ψ によるクエリベクトルの生成を式 (2.41) に示す。

$$\mathbf{q}^1 \cdots \mathbf{q}^N = Q_\psi(\mathbf{s}_{\text{LSTM}}(t-1)) \quad (2.41)$$

クエリベクトル $\mathbf{q}^1 \cdots \mathbf{q}^N$ は、式 (2.42) に示すように、Keys \mathbf{K} と画素毎に内積をとり $\tilde{\mathbf{A}}^n$ を生成する。その後、式 (2.43) に示すように、空間方向に softmax 関数を施すことで attention map \mathbf{A}^n を生成

する.

$$\tilde{A}_{i,j}^n = \sum_l q_l^n \mathbf{K}_{i,j,l} \quad (2.42)$$

$$A_{i,j}^n = \frac{\exp(\tilde{A}_{i,j}^n)}{\sum_{i,j} \exp(\tilde{A}_{i,j}^n)} \quad (2.43)$$

attention map A^n は, Values V のチャンネル方向に沿ってブロードキャストされ, Values V とのアダマール積によって, Answer a^n を生成する. Answer a^n を式 (2.44) に示す.

$$a_c^n = \sum_{i,j} A_{i,j}^n V_{i,j,c} \quad (2.44)$$

式 (2.45) に示すように, Answer a^n を LSTM へ入力し, 時刻 t におけるネットワークの出力である $o(t)$ を生成する.

$$o(t), s_{\text{LSTM}}(t) = \text{LSTM}_\phi(a^1, \dots, a^n, q^1, \dots, q^n, s_{\text{LSTM}}(t-1)) \quad (2.45)$$

この query と attention map を用いることで, 入力画像に対して何を注視しているか (what) と, どこを注視しているか (where) の2つの視点から, エージェントの行動選択を解析を実現している.

2.4.2 エージェントモデルに対する言語的説明

XRL 分野では, エージェントモデルの意思決定に対して, 自然言語によりユーザに説明を提供する言語的説明がある. このエージェントモデルに対する言語的説明手法は, 主にテンプレート文を用いた説明文生成アプローチと, 自由形式での説明文生成アプローチがある.

テンプレート文を用いた説明文生成アプローチとしては, Hayes らの研究 [14] が挙げられる. Hayes らの言語的説明手法は, あらかじめ質問と回答のテンプレート文を人手によって複数用意する. ユーザから提供された質問文を質問テンプレート文にマッピングし, 質問テンプレート文と関連したエージェントの状態に関する要素 (行動や状況など) を抽出する. 抽出した要素を回答テンプレート文に当てはめる形で自然言語により回答文を生成している. ロボティクス分野において, “When do you do xxx?” や, “What do you do when xxx?”, “Why didn ’ t you do xxx?” など質問に回答することが可能であることを報告している.

自由形式での説明文生成アプローチとしては, Ehsan らの研究 [15] と Wang らの研究 [123] が挙げられる. Ehsan らと Wang らは, 人手によって作成された状態と行動ペアとそれに対する説明文のデータセットを用いて, 言語モデルを学習することで, エージェントモデルの意思決定に対する説明文を生成している. Ehsan らは, 状態と行動ペアから説明文への翻訳タスクとして, エージェントモデルの意思決定に対する説明文生成を行う Encoder-Decoder 型の言語モデルを構築している. Wang らは, ビデオゲームタスクを対象とし, 状態であるゲーム画面から重要な特徴を抽出し, それらの

特徴 (敵や障害物など) にもとづいた自然言語による説明文を生成する Encoder-Decoder 型の言語モデルを構築している。これら自由形式での説明文生成アプローチは、テンプレート文を用いた説明文生成アプローチより、表現豊かな説明文を生成することが可能である。

前節で述べたエージェントモデルに対する視覚的説明手法は、画像情報等としてユーザに対して説明を提供する。この手法では、最終的なエージェントモデルの意思決定に対する理解は、ユーザの主観に依存してしまう。一方で、エージェントモデルに対する言語的説明手法は、自然言語での説明文としてユーザに対し説明を提供するため、視覚的説明手法と異なり、エージェントモデルの意思決定を理解する際にユーザの主観に依存しにくい。また、自然言語処理分野では大規模言語モデル (LLM) [16, 17, 18] の登場により、大幅な技術進歩が起きていることから、エージェントモデルの意思決定プロセスに対する言語的説明の実現が期待されている。

第3章

深層強化学習による補助タスクの動的な選択

深層強化学習における学習データは、環境との大規模な試行錯誤を通じて収集するため、学習データに対するサンプル効率がエージェントモデルの性能に大きく影響する。この課題に対し、マルチタスク学習の一種である補助学習を用いた学習効率の向上が取り組まれている。補助学習とは、メインタスクとは異なる目標を持ったタスク(補助タスク)を、エージェントモデルに同時学習することで、メインタスクの学習を促進させるアプローチである。Mirowski らは、3D 環境下でのナビゲーションをメインタスクし、RGB 画像から奥行き情報を予測する depth prediction という補助タスクを用いることで、ナビゲーションの性能向上を報告している [115]。また Kartal らは、エピソードの終端状態に焦点を当て、現状態が終端状態にどれだけ近いかを予測する Terminal Prediction (TP) という補助タスクを提案している [114]。これらをはじめとした深層強化学習における補助学習手法は、特定のメインタスクにおいてエージェントモデルの性能向上が報告されている [113, 114, 115, 116, 13]。しかし、これら補助タスクの有効性はメインタスクに依存し、不適切な補助タスクはメインタスクの学習を妨げる可能性がある。

本章では、エージェントモデルの学習効率向上を目的とし、エージェントモデルのメインタスクに応じて補助タスクを動的に選択するモジュール Auxiliary Selection (AS) を提案する。このモジュールを深層強化学習における教師なし補助タスクを導入した補助学習手法である UNsupervised REinforcement learning and Auxiliary Learning (UNREAL) [13] へ適用することで、様々なメインタスクの効率的な学習を実現する。UNREAL の補助タスクは、メインタスクに依存しない教師なし補助タスクであるため、メインタスクに有効であるか、あらかじめ人手で補助タスクを選定することは、教師あり補助タスクと比較して困難である。そのため、UNREAL に対して AS を適用することで、様々なメインタスクの網羅的かつ効果的な学習を確認する。つまり、本研究ではメインタスクをビデオゲーム攻略とし、補助タスクとして 2.3 節で説明した UNREAL の 3 つの補助タスクを利用する。メインタスクの学習には、UNREAL と同様に Asynchronous Advantage Actor-Critic (A3C) [44] を用いる。AS の有効性を DeepMind lab [144] における 3 つのビデオゲームタスクを用いてゲームスコアにもとづく検証を行う。また、AS がエージェントモデルのメインタスクに対する学習段階に応じて、補助タスクの動的な選択を実現できているか、一定の学習段階ごとに対して詳細に調査する。

本章の構成は次の通りである。3.1 節では、メインタスクに適した補助タスクを動的に選択する AS について述べる。3.2 節では、3.1 節で述べた AS の有効性を確認するため、DeepMind Lab を用いた評価実験について述べる。最後に 3.3 節で本章をまとめる。

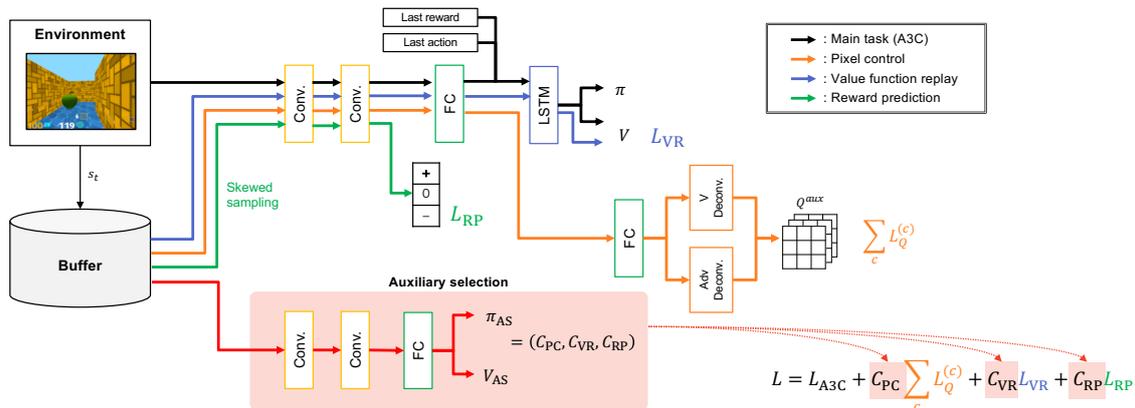


図 3.1: UNREAL + Auxiliary Selection におけるエージェントモデルの概略. ここで Pixel Control, Value function Replay, Reward Prediction は Jaderberg らによって提案された教師なし補助タスクである.

3.1 深層強化学習による補助タスクの選択モジュール

本節では、解くべきビデオゲームタスクに対して最適な補助タスクを選択する Auxiliary Selection (AS) について述べる. 図 3.1 に AS を導入した UNREAL におけるエージェントモデルの概略を示す. AS は、リプレイバッファに保存されたゲーム画面 (RGB 画像) を入力とし、状態価値 $V_{AS}(s)$ と方策 π_{AS} を出力する. AS の方策 π_{AS} は、各補助タスクのバイナリ重みを決定し、この重みは各補助タスクをメインタスクの学習に利用するか否かを示す. 各補助タスクのバイナリ重みは、 $C_{PC} = 0, 1, C_{VR} = 0, 1, C_{RP} = 0, 1$ と定義する. AS の方策 π_{AS} は次式で計算する.

$$\pi_{AS} = (C_{PC}, C_{VR}, C_{RP}). \quad (1)$$

AS は、2つの畳み込み層と1つの全結合層で構成する. 他の補助タスクとは異なり、AS ネットワークはメインタスクネットワークと重みを共有することなく構築され、AS はメインタスクとは独立して学習される. しかし、AS の方策は、メインタスクと同じ報酬にもとづいて学習される. つまり、AS はメインタスクであるビデオゲームのスコアを向上させるために、補助タスクのバイナリ重みを制御する.

3.1.1 損失関数

本手法の損失関数は次のように定式化できる.

$$L = L_{A3C} + C_{PC} \sum_c L_Q^{(c)} + C_{VR} L_{VR} + C_{RP} L_{RP}. \quad (2)$$

ここで、 L_{A3C} はメインタスク (すなわち A3C) の損失であり、 $\sum_c L_Q^{(c)}$, L_{VR} , L_{RP} は、2.3 節で説明した 3 つの教師なし補助タスクの損失である。Pixel Control は、入力画像を $n \times n$ グリッドに分割することで、各グリッドのピクセル値の変位を最大化させる方策をグリッドごとに学習する。つまり、 $L_Q^{(c)}$ は PC の損失であり、グリッド c に対する n ステップ Q 学習の損失である。本手法では、ネットワークパラメータの更新に用いる各補助タスクの損失に、AS で得られたバイナリ重みを乗算することで、補助タスク選択を実現する。式 (2) に示すように、AS のバイナリ重み C_{PC} , C_{VR} , and C_{RP} は本手法の損失関数 L に含まれる。したがって、損失関数 L を用いた AS ネットワークの学習は、AS の出力 C_{VR} , C_{PC} , C_{RP} を含めて損失が 0 に近づくように学習する。つまり、 C_{VR} , C_{PC} , C_{RP} が 0 になるように AS を学習する。このように、AS ネットワークを学習するために異なる損失関数を定義し、メインタスクと補助タスクとは独立して AS を学習する。

AS の損失関数は、状態価値 $V_{AS}(s; \theta_{AS})$ と方策 $\pi_{AS}(a|s; \theta_{AS})$ の損失関数を用いて、次のように定式化できる。

$$L_{ASv} = (r + \gamma V_{AS}(s_{t+1}, \theta_{AS}^-) - V_{AS}(s_t, \theta_{AS}))^2. \quad (3)$$

$$L_{ASp} = -\log(\pi_{AS}(a|s, \theta_{AS}))A(s, a) - \alpha H(\pi_{AS}(\cdot|s, \theta_{AS})). \quad (4)$$

ここで、 θ_{AS}^- は AS ネットワーク更新前のネットワークパラメータ、 r はメインタスクの報酬 (ビデオゲームのスコア) である。また、エントロピー $H(\pi_{AS}(\cdot|s, \theta_{AS}))$ は、ネットワークパラメータが局所解に収束しないように探索を容易にするための項であり、 α はエントロピー $H(\pi_{AS}(\cdot|s, \theta_{AS}))$ のスケールパラメータである。

AS の損失関数は、式 (3), (4) で表される損失の合計として定義される。

$$L_{AS} = L_{ASv} + L_{ASp}. \quad (5)$$

式 (3), (4), (5) から、AS は補助タスクを選択する深層強化学習エージェントであることがわかる。つまり、AS はビデオゲームを解くためのメインタスクモデルとは異なる重みを持ったモデルとして構築され、メインタスクと同じ報酬で最適化することで、ビデオゲームのスコア向上に寄与する補助タスクの選択を実現している。

3.1.2 学習アルゴリズム

本手法の処理の流れを Algorithm 1 に示す。ここで、 θ, θ' はメインタスクモデルのパラメータ、 $\theta_{AS}, \theta'_{AS}$ は AS モデルのパラメータを示す。本手法では、複数の worker による分散学習とモデルパラメータの非同期更新を行う A3C を学習アルゴリズムとして採用する。そのため、各 worker はそれぞれローカルなモデルパラメータ θ', θ'_{AS} とローカルステップ t を持つ。また、グローバルなモデルパラメータ θ, θ_{AS} とグローバルステップ T は worker 間で共有される。

まず、各 worker のモデルパラメータ θ', θ'_{AS} を共有パラメータ θ, θ_{AS} とそれぞれ同期させる。そして、各 worker のエージェントは、ある環境において方策 $\pi(a_t|s_t, \theta')$ に従って、終了条件または t_{max}

Algorithm 1 UNREAL に Auxiliary Selection を導入した場合のアルゴリズム

```
1: //Assume global shared parameters  $\theta$  and  $\theta_{AS}$  and global shared counter  $T = 0$ 
2: //Assume worker-specific parameters  $\theta'$  and  $\theta'_{AS}$ 
3: Initialize local step counter  $t \leftarrow 1$ 
4: repeat
5:   Reset gradients:  $d\theta \leftarrow 0$  and  $d\theta_{AS} \leftarrow 0$ 
6:   Synchronize worker-specific parameters  $\theta' = \theta$  and  $\theta'_{AS} = \theta_{AS}$ 
7:    $t_{start} = t$ 
8:   Get state  $s_t$ 
9:   repeat
10:    Perform action  $a_t$  according to policy  $\pi(a_t|s_t, \theta')$ 
11:    Receive reward  $r_t$  and new state  $s_{t+1}$ 
12:    Store experience  $(s_{t+1}, r_t, a_t)$  in replay buffer
13:     $t \leftarrow t + 1$ 
14:     $T \leftarrow T + 1$ 
15:  until terminal  $s_t$  or  $t - t_{start} == t_{max}$ 
16:  Execute each auxiliary task with the experiences stored in replay buffer
17:  Execute auxiliary selection with the experiences stored in replay buffer
18:  Accumulate gradients  $d\theta$  w.r.t.  $\theta'$ 
19:  Accumulate gradients  $d\theta_{AS}$  w.r.t.  $\theta'_{AS}$ 
20:  Perform asynchronous update of  $\theta$  using  $d\theta$  and of  $\theta_{AS}$  using  $d\theta_{AS}$ 
21: until  $T > T_{max}$ 
```

ステップに達するまで行動を繰り返す。ここで、経験 (s_{t+1}, r_t, a_t) はリプレイバッファに格納される。次に、Auxiliary Selection と 3 つの補助タスクを順番に実行し、式 (2), (5) に示す勾配 $d\theta, d\theta_{AS}$ を計算する。これらの勾配を用いて、グローバルモデルパラメータ θ, θ_{AS} を更新する。この更新は worker ごとに非同期実行され、 T_{max} ステップまで繰り返され、適応的に補助タスクを選択および学習する。

3.2 評価実験

本手法の評価には DeepMind Lab [144] を使用した。DeepMind Lab には、主に `nav_maze_static_01` (maze), `seekavoid_arena_01` (seekavoid), `lt_horseshoe_color` (horseshoe) の 3 ゲームがある。

`maze` は、一人称視点の迷路探索ゲームである。エージェントは、途中でリングを手に入れるごとに +1, ゴールに到達すると +10 を受け取り、一定時間内の獲得得点を競う。エージェントが選択できる行動は、左右への視点移動、前後移動、左右の平行移動の計 6 つである。`seekavoid` は、エージェントがリングを獲得するごとに +1, レモンを獲得するごとに -1, 制限時間内の獲得得点を競うゲームである。エージェントが選択できる行動は、左右への視点移動、前後移動、左右の平行移動の計 6 つである。`horseshoe` は、一人称視点のシューティングゲームです。エージェントは、ステージ上に出現する敵をレーザーで攻撃し、倒すことで +1 を獲得し、制限時間内の獲得得点を競

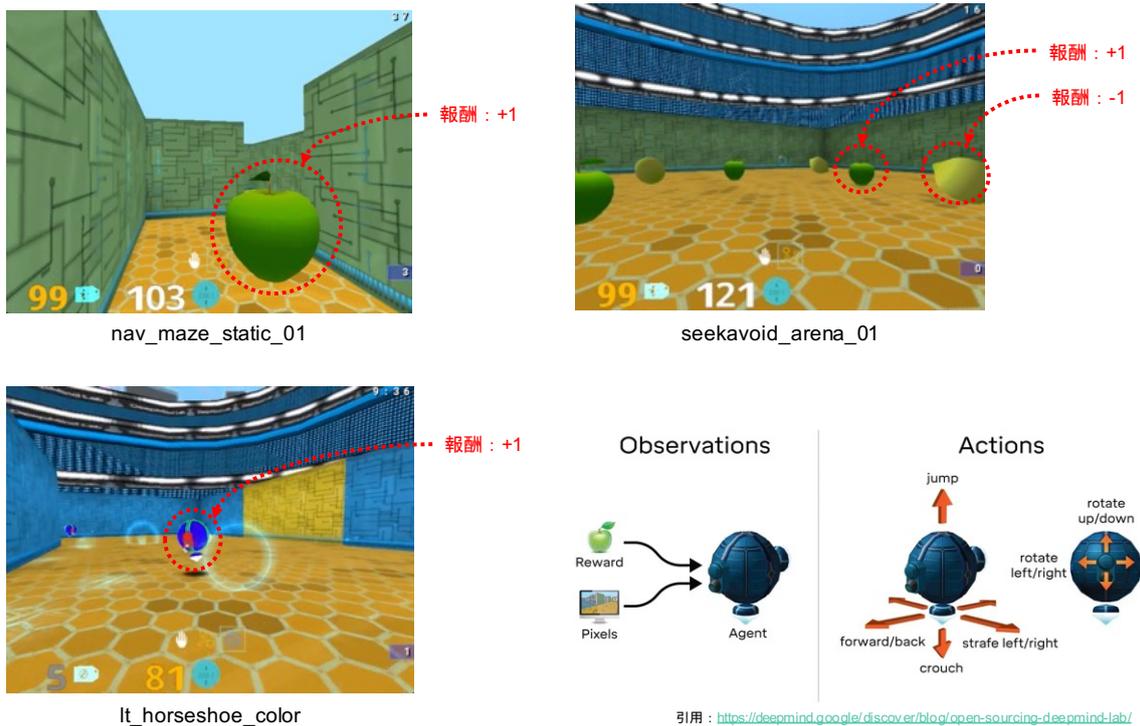


図 3.2: Deepmind Lab における 3 つのゲームとエージェントの観測情報と行動情報. 右下はエージェントの行動を示し, 本実験で用いるゲームタスクでは, jump, crouch, rotate up/down は使用しない.

う. エージェントは, 左右への視点移動, 前後移動, 左右の平行移動, 攻撃の計 7 つである.

本実験では以下のベースラインと性能を比較する.

UNREAL: 学習時に 3 つの補助タスクを使用する.

PC, VR, and RP: 学習時に対象の補助タスクのみを使用する.

それぞれ学習時の共通ハイパーパラメータは統一するものとした. 学習ステップは, maze と seekavoid では 5.0×10^7 ステップ, horseshoe では 1.0×10^8 ステップとした.

3.2.1 ゲームスコアによる性能比較

図 3.3 に, DeepMind Lab における比較手法ごとの学習時のスコア推移を示す. ここで, 横軸はネットワークパラメータを更新するグローバルステップ数を示し, 縦軸は各タスクにおけるスコアを示す.

■ maze

図 3.3 (a) は, maze におけるスコア推移である. UNREAL と PC は, 高いパフォーマンスを達成したが, VR と RP のスコアはほとんど 0 であり, スコアを獲得できていないことが分かる. この結果

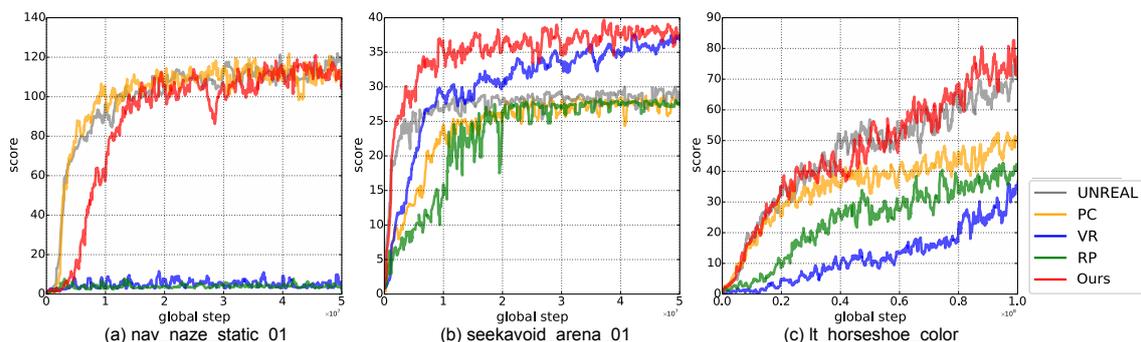


図 3.3: DeepMind Lab の 3 タスクにおける学習時の報酬推移.

は、VR と RP がメインタスクの性能向上に寄与しなかったことを意味する。PC は、入力画像内の画素値を大きく変化させる行動を学習することで、迷路環境におけるエージェントの探索行動を促したと考えられる。これにより、エージェントは迷路環境の隅々まで移動することが可能となった。これらの結果に対し、本手法も UNREAL や PC と同じスコアを獲得できている。

■ seekavoid

図 3.3(b) は,seekavoid におけるスコア推移である。PC は、画素値が大きく変化する行動を学習するという性質上、負の報酬(レモン)を獲得したとしても良い行動と捉えてしまう。この性質は、seekavoid には適していないと考えられる。また、RP は報酬が密集している seekavoid では効率が悪かったと考えられる。一方、UNREAL と VR は高スコアを獲得し、驚くべきことに VR は UNREAL を上回っている。この結果に対し、本手法も VR と同じ性能を獲得できており、VR よりも少ない学習ステップで高い性能を達成している。

■ horseshoe

図 3.3(c) は horseshoe におけるスコア推移である。3つの補助タスクでは PC が最も高いスコアであることが確認できる。これは、horseshoe における敵を倒す行動が画素値を大きく変化させることに繋がるためだと考えられる。一方で、UNREAL はこれら単一の補助タスクのみ用いる場合を上回り、本手法はこの UNREAL と同等の性能を獲得できている。

3.2.2 選択された補助タスクの分析

1 エピソード間の Auxiliary Selection による選択された行動数を図 3.4 に示す。ここで、横軸は Auxiliary Selection により選択された行動 $\{C_{PC}, C_{VR}, C_{RP}\}$ を表し、縦軸はそれぞれの行動が選択された回数を表す。左から、(a) 5.0×10^7 ステップの maze, (b) 5.0×10^7 ステップの seekavoid, (c) 1.0×10^8 ステップの horseshoe である。表 3.1 は、1 エピソードで各補助タスクが選択された回数を示す。補

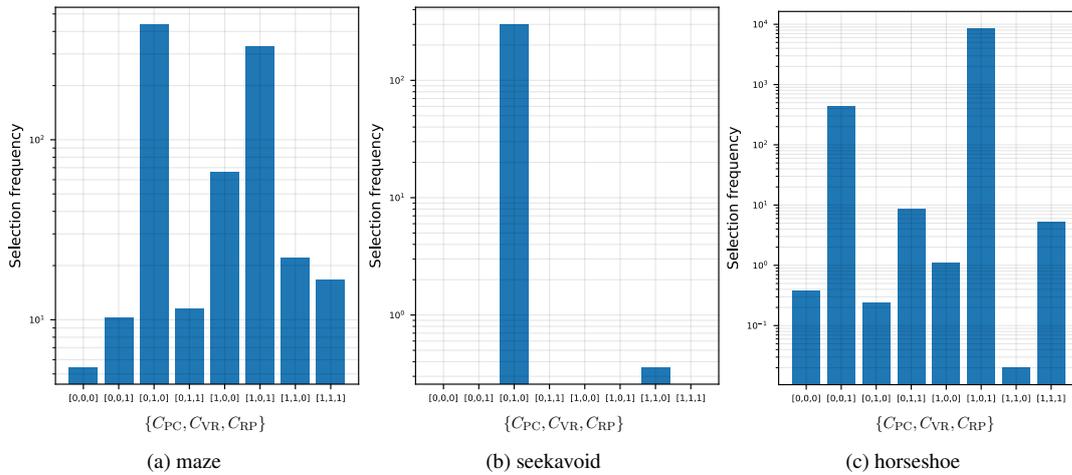


図 3.4: 1 エピソード間における Auxiliary Selection の行動ごとに対する選択回数.

表 3.1: 1 エピソード間における各補助タスクの選択回数. 補助タスクの選択回数は, 50 エピソードの平均値である.

Environment	Auxiliary tasks		
	Pixel Control (PC)	Value functio Replay (VR)	Reward Prediction (RP)
maze	435.4	487.8	369.0
seekvoid	0.3	300.0	0.0
horseshoe	8545.1	14.1	8998.2

助タスクが選択された回数は, 50 エピソードの平均値として算出した. エピソード中の行動ステップ数は, maze が 900, seekavoid が 300, horseshoe が 9,000 である.

maze の結果は, すべての補助タスクが同程度に選択されたことを示している. maze に対する適切な補助タスクは UNREAL または PC であったため, 本手法は全ての補助タスクを同等に選択したと考えられる. seekavoid では, 本手法は VR を安定して選択していることが分かる. この結果は図 3.3 (b) の結果と一致するため, 本手法はメインタスクの学習に寄与する補助タスクのみを選択できていると考えられる. horseshoe では, PC と RP が頻繁に選択されていることが分かる. UNREAL が最高得点を達成したが, horseshoe における Auxiliary Selection では VR が選択されなかった. この理由を分析するため, 次の追加実験を行った.

図 3.3 (c) に示したベースラインに加え, 次のベースラインを追加した: A3C-LSTM (補助タスクなし), PC+RP (Pixel Control と Reward Prediction を使用). 各ベースラインと本手法のスコアを図 3.5 に示す. VR のスコアは A3C-LSTM のスコアより低く, PC+RP は UNREAL や本手法と同じスコアであることがわかる. したがって, 本手法は horseshoe の学習から Value function Replay を取り除くことに成功している.

これらの結果から, Auxiliary Selection がメインタスクの学習に貢献する補助タスクを選択できる

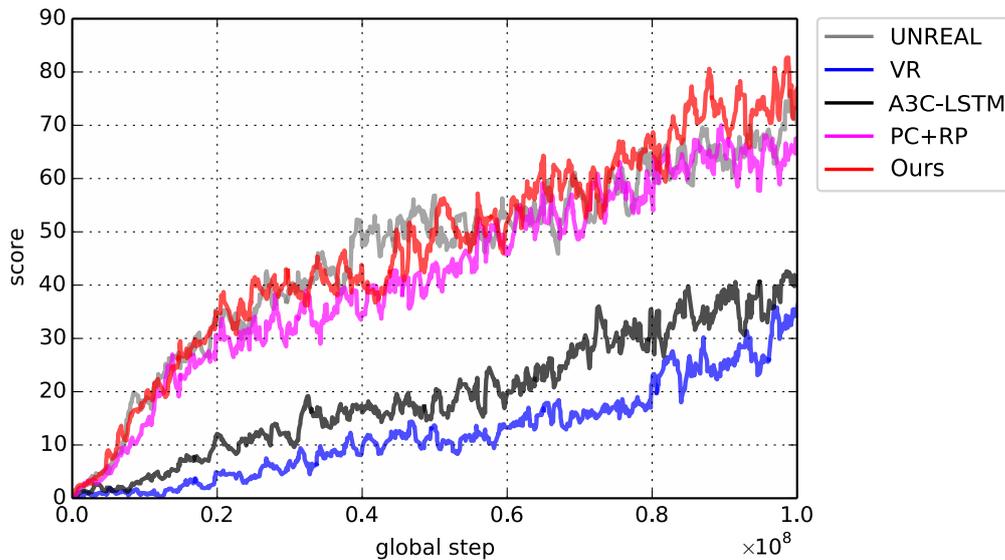


図 3.5: 補助タスクの組み合わせパターンを追加した horseshoe における学習時のスコア推移.

ことを示していることを確認した.

3.2.3 学習段階に応じた補助タスクの動的選択

本節では、学習段階に応じて最適な補助タスクの選択ができているかどうかを確認するために、学習段階ごとにおける補助タスクの選択割合を分析する。seekavoid の各学習段階における 1 エピソード間の補助タスクの選択率を図 3.6 に示す。ここでは、global step 数が 0.2×10^7 , 0.5×10^7 , 1.0×10^7 , 5.0×10^7 の 4 つの学習段階について調査した。つまり、上述の学習ステップ数時における Auxiliary Selection が、1 エピソードあたりに各補助タスクをどの程度選択するかの割合に着目し調査した。この時の補助タスクの選択回数は、50 エピソード間の平均値とした。

図 3.6 から、 0.2×10^7 ステップと 0.5×10^7 ステップでは、全ての補助タスクが同程度に選択されていることがわかる。ここで、UNREAL は 0.2×10^7 と 0.5×10^7 ステップでスコアが約 25 の高スコアを獲得している。これらの結果から、 0.5×10^7 ステップまでの初期学習段階において、本手法は全ての補助タスクを同程度に使用する UNREAL と同様に、各補助タスクを選択していると考えられる。一方で、 1.0×10^7 ステップでは、本手法が VR を優先的に選択していることが確認できる。また、 1.0×10^7 ステップでは UNREAL と VR のスコアが約 30 点と最も高かったことから、本手法は VR が有効な学習段階で VR を選択できていると考えられる。ここで、 5.0×10^7 ステップでは VR のみが選択された。グラフの 1.0×10^7 ステップでは、UNREAL と VR のスコア差は約 7 と大きく、VR が最も高いスコアを獲得している。したがって、Auxiliary Selection により VR の選択回数は、 1.0×10^7 ステップから 5.0×10^7 ステップで増加したと考えられる。これらの結果から、Auxiliary Selection はメインタスクの学習段階に応じて、メインタスクのスコアが最も高くなる補助タスクの組み合わせを選択できることを確認した。

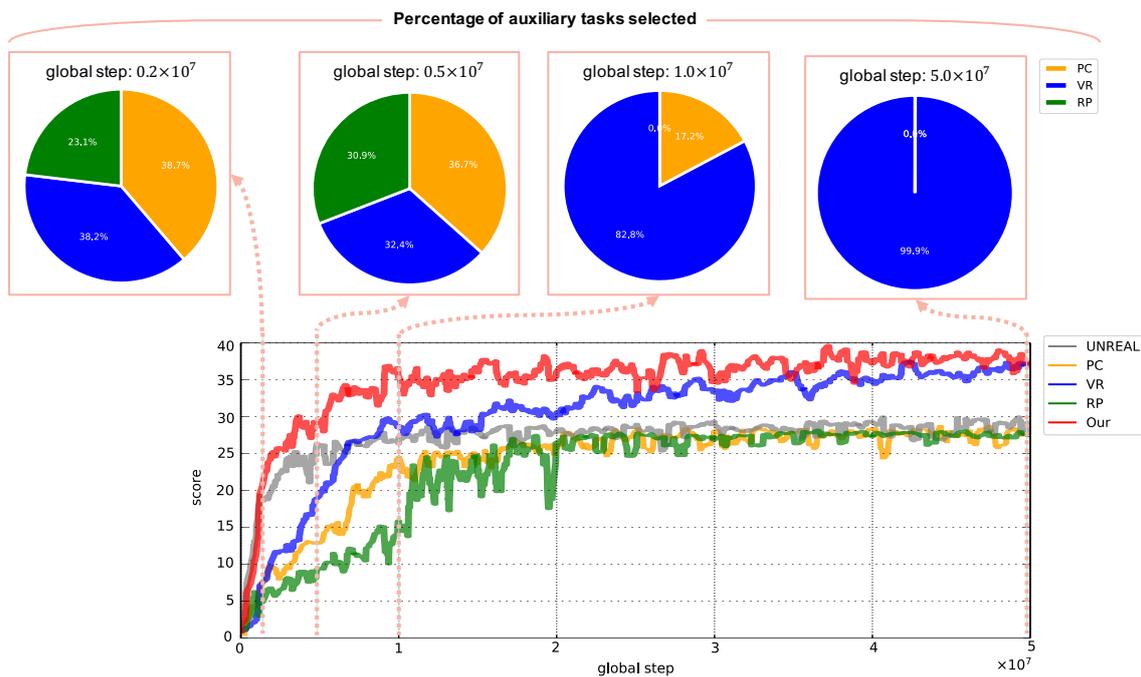


図 3.6: 学習段階に応じた 1 エピソード間における補助タスクの選択率 (seekavoid)。

3.3 まとめ

本章では、メインタスクの学習段階に合わせ、効果的な補助タスクを動的に選択するモジュール Auxiliary Selection (AS) を提案した。補助タスクの選択は、メインタスクに対する性能を向上させるために補助タスクに対するバイナリ重みを制御する深層強化学習エージェントとして設計する。本研究では、ビデオゲーム攻略タスクにおいて教師なし補助タスクを用いる UNREAL に AS を適用し、補助タスクを動的に選択できることを検証した。これにより、メインタスクごとに適した補助タスクを人手で選定する必要がないことを示し、メインタスクに対する効率的な学習促進が可能であることを示した。DeepMind Lab を用いた評価実験から、本手法が各環境（タスク）に最適な補助タスクの組み合わせと同等のスコアを達成することを確認した。また、Auxiliary Selection により選択された補助タスクを詳細に分析することで、本モジュールが解決すべきビデオゲームに適した補助タスクを選択することで、メインタスク（ゲームスコア）の性能向上を示した。また、学習段階ごとに選択された補助タスクを分析することで、本手法はメインタスクの学習段階に応じて最適な補助タスクを選択していることを確認した。今後の課題として、本モジュールを他の補助学習手法に適用し、ビデオゲームタスク以外での検証が挙げられる。

第4章

方策と状態価値に着目した深層強化学習エージェントに対する視覚的説明

2章で述べた通り、深層強化学習エージェントは様々な制御タスクにおいて高い性能を達成している。しかし、これらの深層強化学習エージェントには、ユーザがエージェントモデルの振る舞いを理解することが非常に困難であるというブラックボックス問題がある。この問題には主に2つの要素がある。第1に、学習データがエージェントと環境間のインタラクションを通じて収集されるため、どのようなデータが学習に使われたか、つまりエージェントがどのような経験をもとに行動を獲得したのか不明確である。第2に、エージェントの行動を選択するニューラルネットワーク内部の演算処理が複雑で、選択された行動に対する判断根拠が不明確である。これら2つは、深層強化学習エージェントの実応用に対する大きな障害となっており、エージェントモデルの意思決定を解釈可能にすることは、深層強化学習エージェントの信頼性を担保する上で極めて重要である。

この深層強化学習エージェントに対する説明性/解釈性の向上を目的とした研究分野に、説明可能な強化学習 (eXplainable Reinforcement Learning; XRL) と呼ばれる分野がある (2章参照)。XRLの文脈において、ユーザによる直感的理解が容易なことから、深層強化学習エージェントモデルに対する視覚的説明手法が注目を集めている。視覚的説明とは、入力画像に対してエージェントモデルの注視した領域を画像などの視覚情報を用いて、ユーザに説明を提供するアプローチである。この深層強化学習におけるエージェントモデルに対する代表的な視覚的説明手法としては、摂動を用いるアプローチと、attentionを用いるアプローチが挙げられる。Greydanusらは、エージェントモデルの入力に対し、ガウシアンぼかしにもとづく摂動を付加し、エージェントモデルの出力に対する揺らぎを計測することで、エージェントモデルの方策に対する視覚的な洞察を生成している [4]。また Mottらは、エージェントモデルの方策に対し Query ベースの attention module を導入することで、タスクに関連する重要な情報である2種類の attention (「what」と「where」) を生成している [5]。これらエージェントモデルに対する視覚的説明手法は、深層強化学習の枠組みにおけるエージェントモデルの方策に焦点が当てられている。これは、方策がエージェントの意思決定を表す直接的な出力値であるためである。一方で、状態価値に着目したエージェントモデルの視覚的説明に関する研究は取り組まれていない。深層強化学習における状態価値は、エピソード毎の収益の期待値を表し、方策勾配法にもとづく深層強化学習手法はこの状態価値を最大化するように方策を学習する。つまり、エージェントの意思決定を分析する上で、状態価値は方策と同様に重要であると言える。以上の背景から、actor-critic ベースな深層強化学習手法に着目し、方策と状態価値の2つの観点から深層強化学習エージェントの意思決定を解析可能な XRL 手法 Mask-Attention A3C (Mask A3C) を提案する。

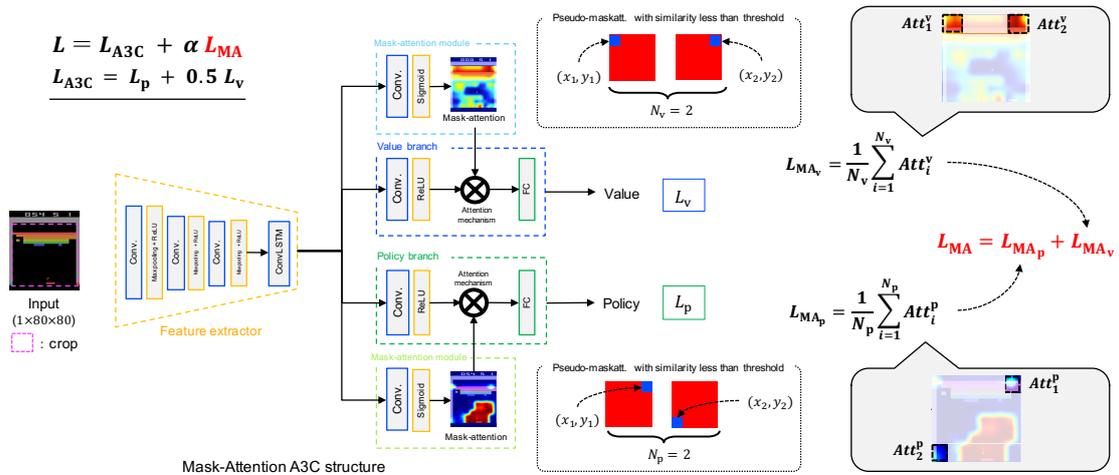


図 4.1: Mask-attention Loss を導入した Mask-attention A3C の概略

本章の構成は次の通りである。4.1 節では、提案手法である Mask-Attention A3C (Mask A3C) について述べる。4.2 節では、Mask A3C に対する網羅的な評価として、Atari2600 のビデオゲームタスクとロボットマニピュレーションタスクにおける評価実験について述べる。最後に 4.3 節で本章をまとめる。

4.1 Mask-Attention A3C

深層強化学習エージェントモデルの意思決定プロセスに対する判断根拠を明らかにする上で、深層強化学習の枠組みにおける方策 (policy) と状態価値 (state-value) という 2 つの観点が重要である。そこで、Actor-Critic ベースな分散型深層強化学習手法である Asynchronous Advantage Actor-Critic (A3C) に着目し、actor と critic のそれぞれに attention 機構を組み込むことで、エージェントモデルの意思決定プロセスを解釈可能とする Mask-Attention A3C (Mask A3C) を提案する。Mask A3C では、Policy branch (actor) と Value branch (critic) に attention 機構を導入する。これにより、各 branch の出力と関連した注視領域を示す mask-attention を獲得する。さらに、エージェントの意思決定に影響を与えない不要な領域を注視しないように制約を課す Mask-attention Loss を導入することで、エージェントの意思決定に対する高い解釈性を有した mask-attention を獲得する。Mask-attention Loss は、mask-attention に特定の摂動を加えることで擬似的な mask-attention を生成し、pesudo mask-attention を用いた際の出力値の変動から注視不要な領域を特定する。Actor と Critic、つまり方策と状態価値の mask-attention を視覚的に解析することで、エージェントの意思決定に対する詳細な理解を実現する。さらに、本手法はエージェントモデルに attention 機構を導入することで、mask-attention を考慮した方策と状態価値の推論を実現し、エージェントの性能向上に貢献する。

4.1.1 Mask A3C モデルの構造概要

Mask A3C のモデル構造は、図 4.1 左に示す。Mask A3C は、Feature extractor, Attention 機構を持つ各 Output branch (方策と状態価値) で構成されている。以下、Mask A3C の各コンポーネントについて詳細に説明する。

Feature extractor. このモジュールは、与えられた状態 s_t から、畳み込み層と再帰型ニューラルネットワーク (RNN) を使用して特徴マップ $F_{fe}(s_t)$ を計算する。時刻 t における状態は s_t と定義され、このタスクにおける s_t は画像情報である。Mnih らは、A3C において LSTM を使用することで入力状態の時系列情報を効率的に考慮でき、エージェントの性能を大幅に向上できることを報告している [44]。しかし、LSTM は入力画像に対する空間情報を考慮できないため、Mask A3C に LSTM を導入する場合、mask-attention の計算が不可能になる。そのため、RNN として時空間情報を考慮できる Convolutional LSTM (ConvLSTM) [143] を採用する。このモジュールで抽出された特徴マップは、Policy branch と Value branch, Mask-attention module に入力される。

Mask-attention module. このモジュールは、方策と状態価値に対する mask-attention を生成する。ここで、value branch に対する mask-attention は $M_v(s_t)$ 、policy branch に対する mask-attention は $M_p(s_t)$ と表現する。mask-attention は、特徴マップ $F_{fe}(s_t)$ に $1 \times 1 \times$ チャンネル数の畳み込み層とシグモイド関数を適用することで生成する。そして生成した mask-attention は、その後それぞれの output branch へ入力される。

Attention 機構を導入した output branch. Policy branch は actor の役割を保有した branch であり方策を出力、value branch は critic の役割を保有した branch であり状態価値を出力する。各 branch の入力、Feature extractor で抽出した特徴マップ $F_{fe}(s_t)$ である。各 branch は $F_{fe}(s_t)$ を受け取り、畳み込み層と ReLU を適用して、それぞれ新しい中間特徴マップ $F_v(s_t)$ と $F_p(s_t)$ を算出する。これらの中間特徴マップと mask-attention は、各 output branch の attention 機構に利用され、各 branch の中間特徴マップに対する mask-attention を使用したマスク処理が施される。つまり、この mask-attention を用いたマスク処理により、最適な方策と状態価値に寄与する特徴マップ上の領域を強調することができる。マスク処理後の各 branch の特徴マップ $F'_v(s_t)$ と $F'_p(s_t)$ は、以下のように計算される。

$$F'_v(s_t) = F_v(s_t) \cdot M_v(s_t), \quad (4.1)$$

$$F'_p(s_t) = F_p(s_t) \cdot M_p(s_t), \quad (4.2)$$

ここで、 $M(s_t)$ は mask-attention である。マスクされた特徴マップ $F'_p(s_t)$ と $F'_v(s_t)$ を全結合層へ入力することで、最終的な出力値である方策と状態価値を算出する。したがって、エージェントはマスクされた特徴マップを用いて、mask-attention によりハイライトされた領域に注目し、現状態に対する行動を選択する。

4.1.2 Mask-attention Loss

エージェントモデルの学習時に Mask-attention Loss を導入し、エージェントの意思決定に影響を与えない不要な領域に対する注視に制限を課す。これにより、出力値である方策と状態価値に寄与する領域のみに着目した mask-attention を得ることができる。これらの不要な領域は、1 ピクセルのみを 0 に設定し、残りピクセルをすべて 1 に設定する疑似的な mask-attention (pseudo-maskatt) を使用して、各 branch の出力値の変動から特定する。ここで、pseudo-maskatt のサイズは mask-attention と同じである。また pseudo-maskatt は、値が 0 のピクセル位置を全ピクセル分走査し、すべての位置に対して作成される。つまり、pseudo-maskatt の数は pseudo-maskatt のサイズ、ピクセル数と同じである。Mask-attention Loss の計算例を図 4.1 右に示す。

Mask-attention Loss の計算フローは以下の通りである。

1. 環境から観測した状態を入力として、各 branch における mask-attention と出力値 (方策または状態価値) を算出する。
2. mask-attention module (attention 機構の attention weight) に pseudo-maskatt を利用した場合の方策と状態価値を算出する。ここで、エージェントモデルへの入力値は 1 と同じである。
3. 1 と 2 の出力値 (方策と状態価値) の差分値を算出する。方策の差分値は、方策が確率分布であるため、Kullback-Leibler (KL) divergence を用いて算出する。状態価値の差分値は、L1 ノルムを用いて算出する。つまり、各差分値 Dif_p, Dif_v は、以下の式で計算される。

$$Dif_p = |\text{KL}(\pi(\mathbf{s}_t, M_p(\mathbf{s}_t)) \parallel \pi(\mathbf{s}_t, M_{\text{pseudo}}))|_1, \quad (4.3)$$

$$Dif_v = |V(\mathbf{s}_t, M_v(\mathbf{s}_t)) - V(\mathbf{s}_t, M_{\text{pseudo}})|_1, \quad (4.4)$$

ここで、 \mathbf{s}_t は時刻 t における入力状態、 $M(\mathbf{s}_t)$ は mask-attention、 $\pi(\mathbf{s}_t, \cdot)$ は方策、 $V(\mathbf{s}_t, \cdot)$ は状態価値、そして $\text{KL}(\cdot \parallel \cdot)$ は KL divergence である。Dif の値が低いほど、mask-attention と pseudo-maskatt の出力値の変化が少ないことを意味する。つまり、Dif が低い場合は、pseudo-maskatt におけるピクセル値 0 に対応する領域は、エージェントの意思決定プロセスに影響を与えない不要な領域であることを意味する。

4. 閾値処理によって出力値の差異 Dif の少ない pseudo-maskatt を特定する。
5. 特定した pseudo-maskatt のピクセル値 0 に対応する mask-attention の値を深層強化学習モデルが注視しないように制限する Mask-attention Loss を計算する。Mask-attention Loss L_{MA} は以下のように計算される。

$$L_{MA} = L_{MA_p} + L_{MA_v}, \quad (4.5)$$

$$L_{MA_p} = \frac{1}{N_p} \sum_{i=0}^{N_p} Att_i^p, \quad (4.6)$$

$$L_{MA_v} = \frac{1}{N_v} \sum_{i=0}^{N_v} Att_i^v, \quad (4.7)$$

ここで、 Att^p, Att^v は、policy branch と value branch に対する pseudo-maskatt のピクセル値 0 に対応した mask-attention の値である。また、 N_p, N_v は Att^p, Att^v の数である。

6. その他の損失関数は A3C と同様である。そのため、5 で計算した Mask-attention Loss を A3C の損失関数に加算し学習を行う。Mask-attention Loss を加算する際は、A3C の損失関数とのバランスを考慮するため、学習率 α を乗算し、損失のスケールを調整する。この学習率はハイパーパラメータである。

この損失を学習プロセスに導入すること、mask-attention における出力値に寄与しない領域の値を 0 に近似する。つまり、Mask-attention Loss は、mask-attention をエージェントモデルの方策/状態価値の算出に寄与する領域のみを示したマップとなるように学習を促す効果がある。これにより、mask-attention の解釈性が向上し、エージェントモデルの振る舞いに対するユーザの理解が促進するを実現する。

4.2 評価実験

本節では、ビデオゲーム攻略タスクを用いた実験により、Mask-attention A3C (Mask A3C) と Mask-attention Loss の有効性を評価する。ビデオゲームで確認した効果を、深層強化学習の応用が成功しているロボットマニピュレーションタスクにおいても同様に有効か確認する。また、mask-attention がユーザにとって高い解釈性を持つことを確認するために、mask-attention を用いたエージェントモデルの行動予測に関するアンケート調査をユーザを対象に実施する。4.2.1 - 4.2.7 節はビデオゲーム攻略タスクにおける評価実験、4.2.8 節はロボットマニピュレーションタスクにおける評価実験について述べる。4.2.9 節は、ユーザがエージェントの意思決定を理解するための mask-attention の解釈性に関するアンケート調査について述べる。

4.2.1 ビデオゲーム攻略タスクにおける実験詳細

OpenAI gym のビデオゲームタスク [145] を用いたエージェントモデルの意思決定に対する解析と、Mask A3C による性能向上の効果に対する評価実験について述べる。使用したビデオゲーム、Breakout (BO), Ms. Pac-Man (MP), Seaquest (SQ), Space Invaders (SI), Beamrider (BR), Fishing Derby (FD) の 6 ゲームである。

表 4.1 に、本実験における実験詳細を示す。ここで比較手法の Policy Mask A3C と Value Mask A3C は、片側の branch にのみ、つまり policy branch または value branch のどちらかに attention 機構を実装した Mask A3C であり、Mask A3C MaskattL は Mask-attention Loss を適用した Mask A3C のことである。Mask A3C MaskattL における Mask-attention Loss の適用タイミングとして、 0.8×10^8 ステップ以降での導入とした。つまり、global step 数が 0.8×10^8 ステップになるまでは、A3C と Mask A3C と同一の損失関数にて学習を行う。学習初期段階のエージェントモデルは学習が不十分であるため、どこの領域が出力にとって重要か定まっていない。そのため、エージェントモデルの注視領域

表 4.1: ビデオゲーム攻略タスクにおける実験条件

Comparison methods	A3C, Mask A3C, Policy Mask A3C, Value Mask A3C, and Mask A3C MaskattL	
Training conditions	number of worker	30
	optimizer	Adam
	global steps	1.0×10^8
	learning rate	0.0001
	discount rate	0.99
	termination condition of an episode	reached 1.0×10^4 step or the end of 1 game
	skip frame	4
	threshold value for select pseudo-maskatt	0.1
	start step of Mask-attention Loss	0.8×10^8

が明確になった後に、不要な注視領域の除外を促進させるために、学習後期段階での Mask-attention Loss の導入とした。また Mask-attention Loss の計算における差分値が小さい pseudo-maskatt を特定するための閾値は 0.1 とした。

本実験では、以下の 5 つの指標をもとに評価する。

- Mask-attention Loss が mask-attention に与える影響
- 従来手法を用いた注視領域可視化による定性的比較
- mask-attention の可視化によるエージェントモデルの意思決定プロセスに対する分析
- Atari 2600 におけるスコア比較
- inverse mask-attention を用いたスコア減少による比較
- mask-attention に着目した新しい状態に対するエージェントの反応

エージェントモデルの実装詳細。 ビデオゲーム攻略タスクの実験で用いるエージェントモデルの詳細を説明する。入力にはゲーム画面のグレースケール画像、出力は各ゲームの現状態に対する制御コマンドである。入力のグレースケール画像は、 80×80 にリサイズする。Feature extractor は、3 つの畳み込み層 (畳み込み処理後に maxpooling と ReLU を適用) と ConvLSTM で構成する。畳み込み層は出力次元数 32 の層が 2 層、出力次元数 64 の層が 1 層である。また、ConvLSTM の隠れ状態の出力次元数は 64 である。Policy branch は、畳み込み層 (畳み込み処理後に ReLU を適用) が 1 層、全結合層が 1 層、softmax 関数にて構成する。ここで、各層は出力次元が 32 の畳み込み層と、エージェントの行動数と同じ出力ユニット数を持つ全結合層である。Value branch は、出力次元が 32 の畳み込み層 (畳み込み処理後に ReLU を適用) が 1 層、出力ユニット数が 1 の全結合層が 1 層にて構成す

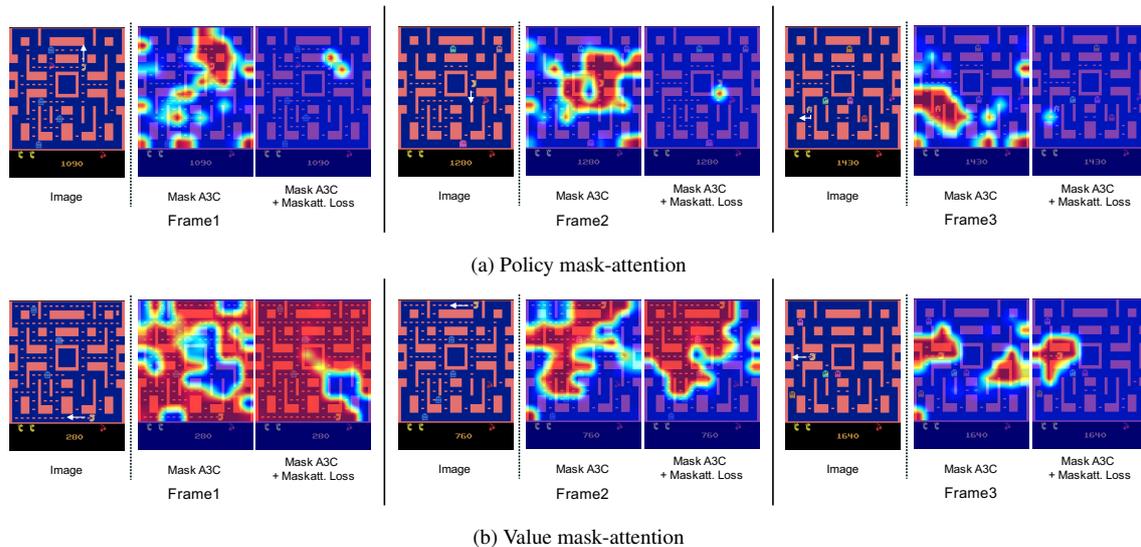


図 4.2: Mask-attention Loss による mask-attention の比較. 矢印はパックマンの進行方向を示す.

る. 本実験における A3C は, attention 機構 (mask-attention module) を除いた Mask A3C と同じモデル構造となっている.

4.2.2 mask-attention に対する Mask-attention Loss の効果

図 4.2 は, Ms. Pac-Man (MP) における同一フレームでの Mask A3C と Mask A3C MaskattL の mask-attention 可視化例を示す. (a) から, Mask A3C MaskattL が Mask A3C と比較してパックマンの進行方向を強く注視していることが確認できる. MP におけるエージェントの制御は, 制御対象であるパックマンをどちらの方向に移動するかを制御することである. そのため, Mask A3C MaskattL ではパックマンの進行方向を強く注視していると考えられる. (b) から, Mask A3C と Mask A3C MaskattL の両方がスコアの源であるクッキーを注視している. また Mask A3C MaskattL は, Mask A3C と比較して画面上にある残りのクッキーを網羅的に注視できていることが確認できる. これらの結果から, Mask-attention Loss を導入することで, 方策と状態価値の両方に対して注視領域が限定されていることが分かる. また, 注視領域が限定されることで, 特定の領域のみを強調した mask-attention を獲得できる. そのため, ユーザにとってエージェントの意思決定に対して解釈性の高い mask-attention を獲得可能とした

4.2.3 従来手法を用いた注視領域可視化による定性的比較

mask-attention が有用であることを, エージェントモデルに対する視覚的説明の従来研究と比較する. また, エージェントモデルのパラメータ数と注視領域の計算時間を検証することで, 本手法の注視領域に関する計算コストを確認する. ここで, エージェントモデルに対する視覚的説明の従来

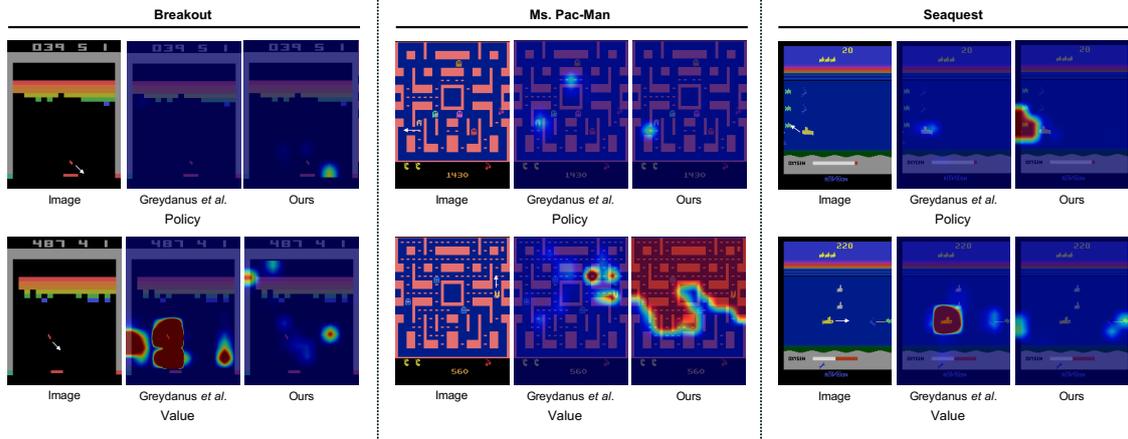


図 4.3: エージェントモデルの注視領域可視化に関する Greydanus らの手法との比較. ここで, Ours は Mask A3C MaskAttL による mask-attention の可視化例である.

手法として, 摂動にもとづく Greydanus らの手法 [4] を用いる. Greydanus らの手法は, エージェントの行動選択を解釈するための顕著性マップを生成する顕著性計算手法であり, 入力画像の特定のピクセルに摂動を付加することで摂動画像を生成する. 顕著性マップは, これらの摂動画像をエージェントモデルの入力として使用した際の出力値の変動にもとづいて, 摂動が付加されたピクセルの重要度から計算される.

図 4.3 は, Greydanus らの手法と Mask A3C (Ours) における注視領域の可視化例を示している. Greydanus らの手法は, Breakout のボール, Ms. Pac-Man のパックマン, Seaquest の自潜水艦などの特徴的なオブジェクトを注視したマップである. 一方で Ours は, ボール, パックマンの移動方向, 潜水艦の攻撃対象となる魚など, エージェントの行動により影響を受ける対象や領域を注視していることが確認できる. さらに Ours と比較すると, Greydanus らの手法では, 注視領域のないマップ (Breakout における policy) や, エージェントの行動と無関係な領域を注視するマップ (Ms. Pac-Man における policy) が生成されていることが分かる.

表 4.2 は, 各手法における注視領域の計算時間とエージェントモデルのパラメータ数を示す. ここで, A3C は視覚説明手法を適用しないエージェントモデルであり, それぞれの手法に対する基準となる値である. Mask A3C は, 方策と状態価値の推論と同時に mask-attention を計算する mask-attention module を追加しているため, 計算時間とモデルパラメータを増加させると考えられる. しかし, mask-attention module は, 1×1 畳み込み層と Sigmoid 関数で構成されるシンプルな構造であるため, 方策と状態価値の推論や注視領域の計算をリアルタイムに実行することが可能である. 一方, Greydanus らの手法は, エージェントモデルの構造に対するアプローチではないため, モデルパラメータは A3C と比較して同じ値である. しかし, この手法は, 摂動画像を用いて注視領域を計算するため, 計算時間が大幅に増加する.

これらの結果から, mask-attention module というシンプルな構造を導入することで, Greydanus らの手法と比較して注視領域の計算時間を大幅に削減し, エージェントの行動選択に関連する領域を示した注視領域の可視化を実現した.

表 4.2: Breakout における 1 フレームあたりに対する注視領域の計算時間とエージェントモデルサイズ. ここで A3C の計算時間は, 方策と状態価値を計算する推論時間を示す. 計算時間の計測には, NVIDIA RTX A6000 を使用した.

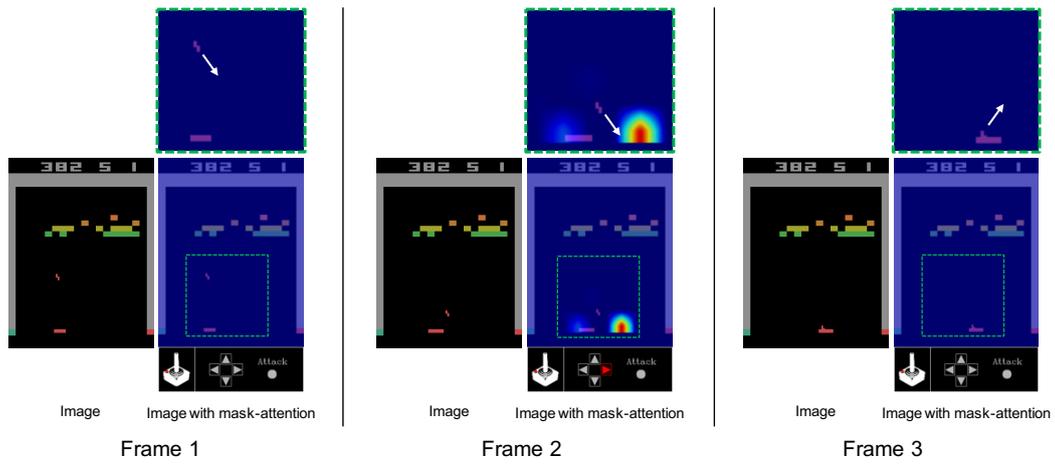
method	calculate time [second]	model parameters
A3C [44]	7.46×10^{-4}	343.909k
Greydanus <i>et al.</i> [4]	3.18×10^{-1}	343.909k
Ours	9.31×10^{-4}	344.039k

4.2.4 mask-attention の可視化によるエージェントモデルの意思決定プロセスに対する分析

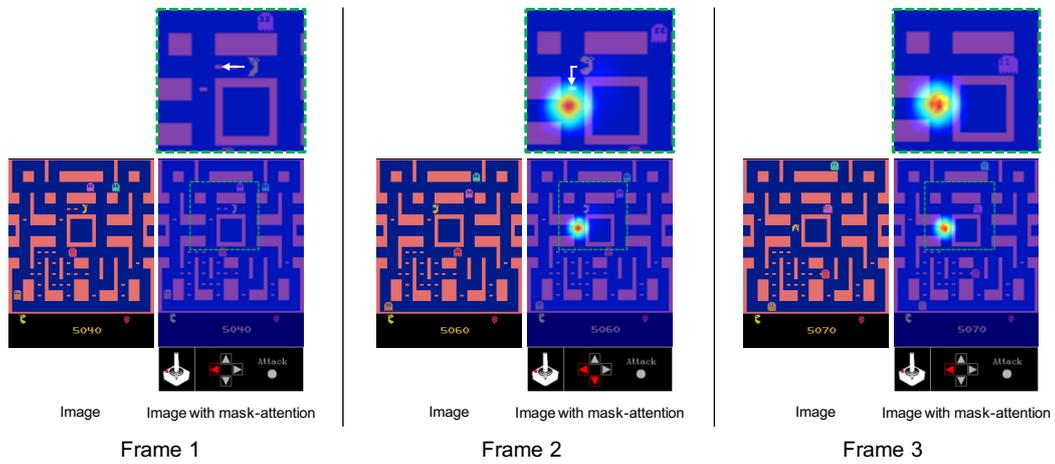
図 4.4, 4.5 は, Mask A3C MaskattL による Breakout, Ms. Pac-Man, Seaquest における mask-attention の可視化例を示す. 以下では, これらの図に示す通り, mask-attention を用いたエージェントモデルの意思決定について解析する.

■ Breakout (BO)

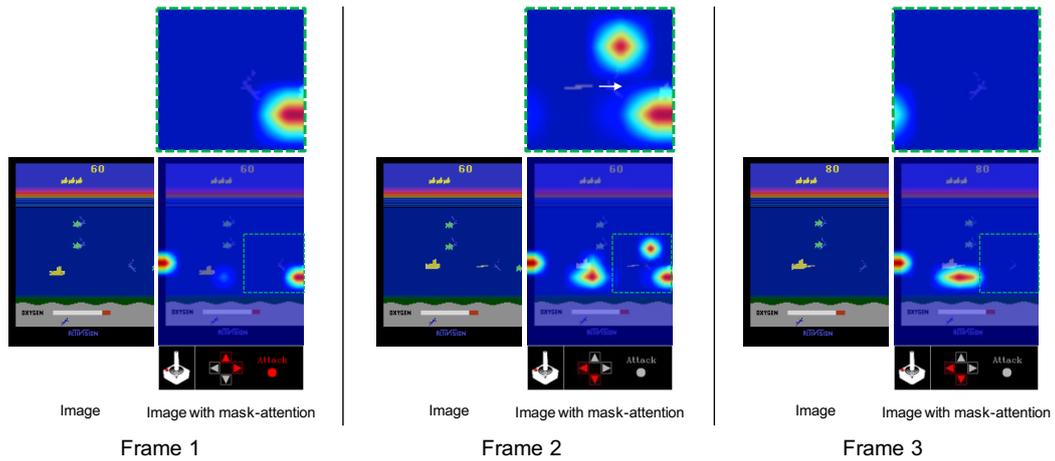
BO は, プレイヤーがパドルを操作してボールを打ち返し, 画面上部のブロックを破壊するゲームである. エージェント (制御対象はパドル) の行動は, No-op, Left, Right の 3 つである. 図 4.4a は, BO における方策の mask-attention を示している. Frame 1 では, ボールはパドルの右側に向かって移動している. Frame 2 では, ボールがパドルに近づくと, エージェントモデルはボールの進行方向を注視し, Right という行動を選択している. Frame 3 では, Frame 2 で注視した領域にパドルが移動し, ボールを打ち返していることが確認できる. したがって, エージェントはボールの進行方向を予測し, そのボールを打ち返すようにパドルを制御していると考えられる. 図 4.5a は, BO における状態価値の mask-attention を示している. Point 1 では, エージェントは注視領域を示しておらず, グラフ上の状態価値も低い値である. Point 2 では, エージェントはブロック左端を注視しており, その時の状態価値はグラフから高いことが分かる. ここで, Point 2 で注視しているブロック左端はブロック数が少ないことが確認できる. ブロック上部のスペースにボールを跳ね返すと, ブロック上部のスペース内でボールがバウンドを繰り返し, 多くのブロックを短時間で破壊することができる. そのため, この行動は Breakout において高得点を得るための重要な戦略の 1 つであると言える. これらの結果から, エージェントがブロック上部のスペースにボールを到達させることの重要性を認識していると考えられる.



(a) Breakout. 矢印はボールの進行方向を示している.

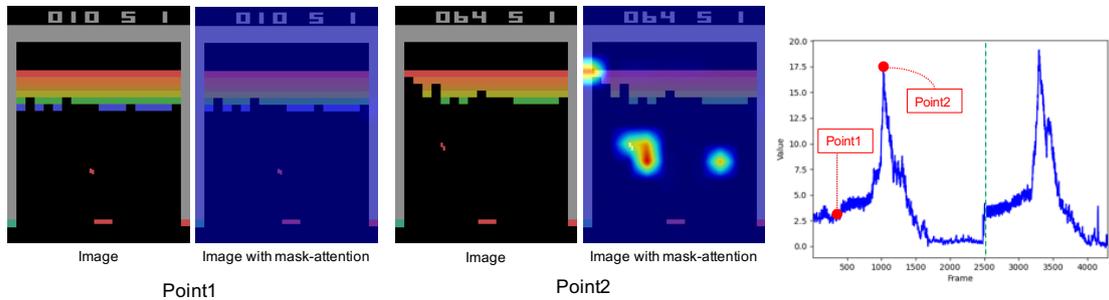


(b) Ms. Pac-Man. 矢印はパックマンの進行方向を示す.

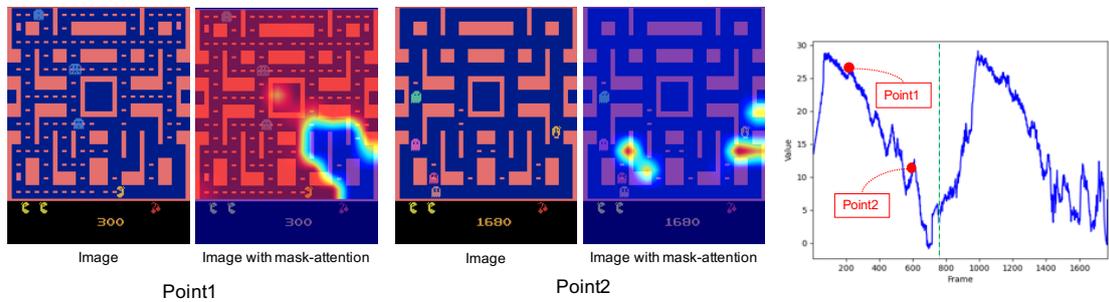


(c) Seaquest. Frame 2 の矢印は潜水艦を攻撃している魚雷の進行方向を示す.

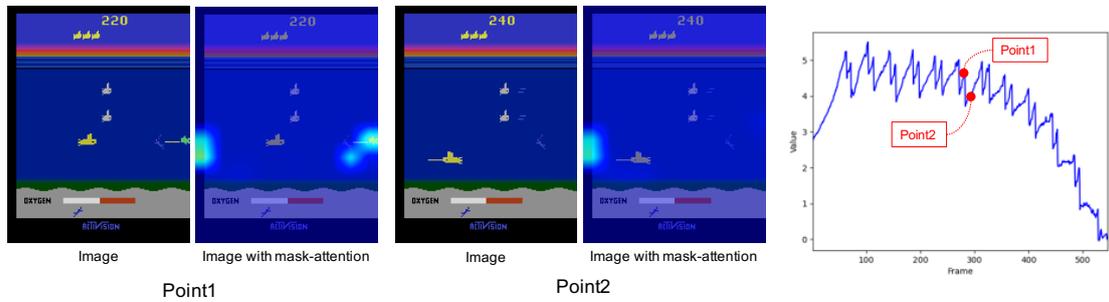
図 4.4: Policy branch における mask-attention の可視化例. 可視化結果は, Mask A3C MaskAttL を用いた mask-attention の可視化例である. Image with mask-attention 下部のコントローラは, 現状態で深層強化学習エージェントが選択した行動を示す.



(a) Breakout



(b) Ms. Pac-Man



(c) Seaquest

図 4.5: Value branch における mask-attention の可視化例。画像は、Mask A3C MaskattL を用いた mask-attention の可視化例である。状態価値が大きく変化する 2 フレームでの mask-attention の例を示す。右のグラフは状態価値の遷移を示し、グラフ中の緑破線は各ゲームにおける次ステージへの遷移を示す。

■ Ms. Pac-Man (MP)

MP は、プレイヤーを操作して、敵を避けながら散らばったクッキーを集めるゲームである。エージェント (制御対象はパックマン) の行動は、No-op, Up, Down, Left, Right, Up+Left, Up+Right, Down+Left, Down+Right の 9 つである。図 4.4b は、MP における方策に対する mask-attention を示している。Frame 1 では、エージェントが Left を選択し、パックマンが左に移動する。Frame 2 では、パックマンが十字路に到達し、エージェントは Down+Left を選択し、パックマンの下にあるクッキーを注視している。Frame 3 では、パックマンは Frame 2 でエージェントが注視していた領域に移動し、クッキーを獲得していることが確認できる。このように、エージェントは制御対象がパックマンであることを認識し、パックマンをクッキーに向かって移動するように制御していると考えられる。図 4.5b は、MP における状態価値に対する mask-attention を示している。Point 1 では、ゲームが開始した直後のため、エージェントは画面全体を注視している。一方、Point 2 では、画面上のクッキー数が減ってきたため、注視領域が縮小している。また、Point 1 から Point 2 にかけて、画面上のクッキー数が減ったため、状態価値も減少していることが確認できる。これらの結果から、エージェントがクッキーを得点源として認識していると考えられる。実際に、MP ではクッキーを収集することが目的であるため、これらは正しい認識であると言える。

■ Seaquest (SQ)

SQ は、プレイヤーが潜水艦を操作してダイバーを救出しながら、敵潜水艦や魚を撃破するゲームである。エージェント (制御対象は潜水艦) の行動は、No-op, Up, Down, Left, Right, Attack の 6 つである。図 4.4c は、SQ における方策に対する mask-attention を示している。Frame 1 では、エージェントは画面右から現れた魚を注視しており、ここで Attack を選択している。Frame 2 では、エージェントの攻撃であるミサイルが Frame 1 で注視していた魚に向かっていく。Frame 3 では、エージェントは魚を注視しておらず、Frame 1 で注視していた魚を撃破していることが確認できる。これらの結果は、魚が現れると同時にエージェントは魚を認識し、その魚を倒すために自潜水艦を制御していると考えられる。図 4.5c は、SQ における状態価値に対する mask-attention を示している。魚を破壊する直前の Point 1 では、エージェントは魚に対して注視している。Point 2 では、Point 1 で注視していた魚が撃破され、魚に対する注視が消えていると同時に、状態価値が減少している。これらの結果から、エージェントは魚を撃破することが SQ にとって重要であると認識していると考えられる。実際に、SQ ではダイバーの救助と同時に敵 (敵潜水艦と魚) の撃破が目的であるため、これは正しい認識であると言える。

■ Atari2600 における mask-attention によるエージェントモデルの視覚的説明に関するまとめと考察

Actor-Critic ベースな深層強化学習手法の output branch に attention 機構を実装することで、2 種類の異なる観点 (方策と状態価値) に対する mask-attention が獲得できることを確認した。方策の mask-

attention は、エージェントが選択した行動に寄与する領域を直接的に示すことを確認した。これは、方策が現状態におけるエージェントの選択行動に対する確率分布を表すためである。一方、状態価値の mask-attention は、ビデオゲームの目的を示す特徴的なオブジェクトや領域を示していることを確認した。これは、状態価値が現状態におけるリターンの期待値を表しているためである。ここで、リターンとはエピソードにおける報酬の合計である。これらの2つの観点に着目した mask-attention を可視化することで、ビデオゲームタスクにおいてエージェントの意思決定を明らかにできることを示した。

4.2.5 Atari 2600 におけるスコア比較

Atari 2600 ゲームにおける各比較方法の 100 エピソード間の最大スコアと平均スコアを表 4.3 に示す。BO における平均スコアでは、A3C が最高得点を獲得していることが確認できるが、最大スコアでは、Mask A3C が全ての手法において BO で可能な最高得点 (864) を獲得している。BO は外部要因のないシンプルなゲーム (ボールとブロックのみが影響を受ける) であるため、Mask A3C は A3C よりも大幅に低いスコアを獲得することがなかったと考えられる。つまり、BO は A3C や Mask A3C で十分に学習可能なビデオゲームタスクのため、優位な差は現れなかったと考えられる。一方で、MP, SI, FD では、policy branch に mask-attention を導入することで、A3C よりも高いスコアを達成していることが確認できる。これは、mask-attention がエージェントの行動選択に影響するオブジェクト (MP ではクッキーや敵、SI では防御壁やインバーダー、FD ではプレイヤーに最も近い魚) を、効果的に強調できるためだと考えられる。BR では、いずれの手法でもスコアに大きな差異は確認できない。BR は、倒すと得点になる敵と、避けなければならない敵が存在し、これら敵の外見は非常に似ている。mask-attention はあくまで入力画像および特徴マップに対する強調のため、mask-attention による強調のみでは、敵の詳細な特性の違いを捉えるには不十分であったと考えられる。同様に、SQ においてもいずれの方法でもスコアに大きな差異は確認できない。SQ は、エージェントが魚を攻撃する行動や酸素を補給する行動を取得できるかどうかによって、獲得スコアは大きく変動する。4.2.4 節の mask-attention の可視化例で示したように、エージェントは魚を撃破するような行動を選択しているが、画面下部の酸素ゲージを注視していないことが確認できる。可視化結果からも分かる通り、mask-attention のみでは、エージェントに酸素ゲージに注視させるように誘導できないため、いずれの方法でもスコアに大きな差は出なかったと考えられる。Mask-attention Loss を用いた手法の平均スコア (BO, MP, SQ, BR) は、Mask A3C と比較し低下していることが確認できる。一方で、SI と FD については A3C と比較して、スコア向上または同程度のスコアを達成している。これらの結果から、Mask-attention Loss は注視領域を限定し、注視対象の周辺情報を考慮することが難しくなるものの、A3C と比較してスコアが向上し、Mask A3C よりも高い解釈性を持つ mask-attention を実現できたと考えられる。

表 4.3: Atari 2600 における 100 エピソード間の最大/平均スコア。ここで、各手法における 5 試行のうち、平均スコアが最も高かったモデルを用いた。また、太字は各ゲームにおける最大/平均スコアの最高スコアを示す。

Attention mechanism		Mask-attention		BO		MP		SQ		SI		BR		FD	
Policy	Value	max	loss	max	mean	max	mean	max	mean	max	mean	max	mean	max	mean
✓		864	662.0	5380	4573.3	2760	2728.2	19505	18531.8	34748	28341.1	41	32.1		
		864	595.8	6330	4833.8	2820	2784.0	19860	19102.8	32604	28495.3	41	37.5		
	✓	864	606.9	4830	4044.5	2820	2786.4	19675	18537.8	35108	28205.7	43	36.1		
✓	✓	864	650.5	9750	6716.3	2780	2748.6	20510	19508.1	31440	26113.1	47	39.3		
✓	✓	864	573.7	7140	5681.9	1860	1835.2	20700	19099.3	31200	24685.1	63	39.3		

表 4.4: Policy mask-attention の注視領域を反転させた場合とさせなかった場合のスコア比較。ここで、inverse attention の ✓ は、mask-attention の注視領域を反転させるか否かを示す。また random はランダムな行動選択によるスコアである。

Attention mechanism		Mask-attention		Inverse attention		BO		MP		SQ		SI		BR		FD	
Policy	Value	max	loss	max	mean	max	mean	max	mean	max	mean	max	mean	max	mean	max	mean
✓		864	595.8	6630	4833.8	2820	2784.0	19860	19102.8	32604	28495.3	41	37.5				
		4	2.2	290	268.9	280	158.2	805	306.9	4996	1554.2	-49	-75.7				
✓	✓	864	650.5	9750	6716.3	2780	2748.6	20510	19508.1	31440	26113.1	47	39.3				
		18	5.2	810	446.0	220	160.6	535	311.4	2216	1156.8	-57	-84.4				
✓	✓	864	573.7	7140	5681.9	1860	1835.2	20700	19099.3	31200	24685.1	63	39.3				
		2	0.9	1080	671.5	520	222.0	1150	488.6	660	232.8	-43	-63.4				
		5	1.2	1080	247.8	300	82.8	460	142.1	852	356.5	-85	-93.1				
		random															

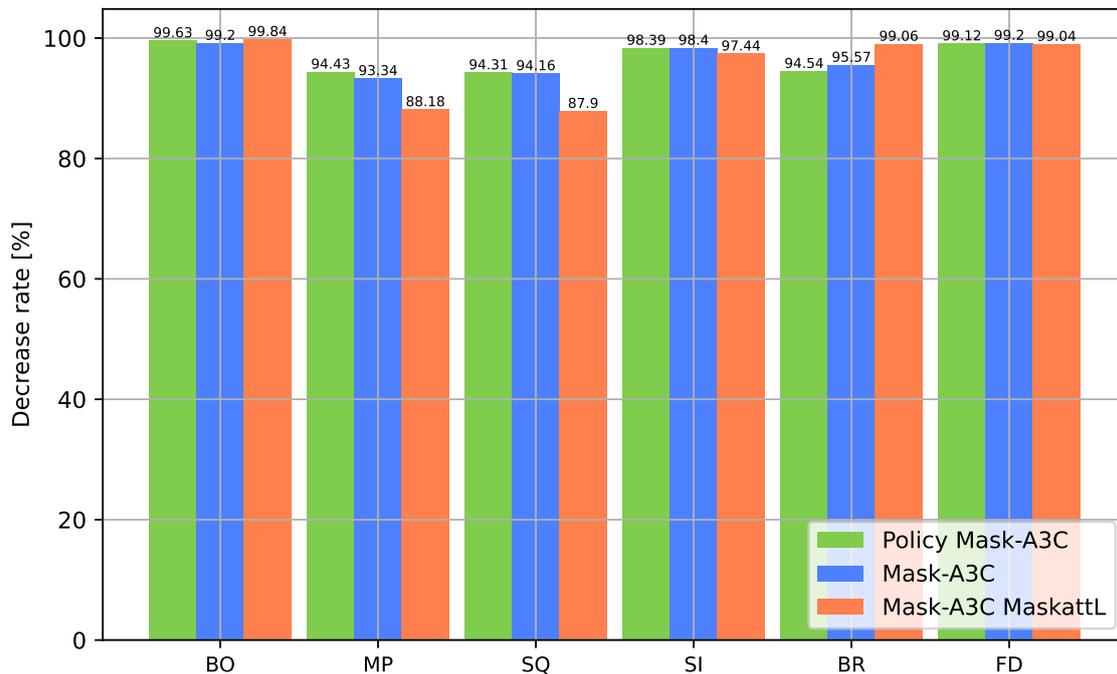


図 4.6: inverse policy mask-attention による平均スコアの低下率. ここで, 平均スコアは 100 エピソード間における平均スコアである.

4.2.6 Inverse mask-attention を用いたスコア減少による比較

mask-attention がエージェントの意思決定プロセスに対する注視領域を示しているかどうか調査する. 本実験では, policy branch の mask-attention に着目し, エージェントの行動選択に対する寄与を確認する. mask-attention がエージェントの意思決定に対する判断根拠を示すと仮定し, mask-attention の注視領域を反転 (値の高い領域を低く, 低い領域を高く) した際のゲームスコアの変動から評価した. つまり, mask-attention を反転させてもエージェントモデルが獲得するゲームスコアに変動がない場合は, mask-attention がエージェントの行動選択に影響を与えていないことを示し, mask-attention はエージェントの行動選択に寄与していないこと意味する. 一方, ゲームスコアが減少する場合は, mask-attention がエージェントの行動選択に影響を与える領域を示し, mask-attention はエージェントの行動選択に大きく寄与していることを意味する. この手法では, 学習済みエージェントモデルの policy branch の mask-attention を反転することで, inverse mask-attention を作成し, このマップを policy branch の attention 機構に用いることで行動選択を実行する. mask-attention の反転の有無によるゲームスコアを比較することで, mask-attention がエージェントの意思決定に対する視覚的説明として有効であるかを確認する. mask-attention $M(\cdot)$ の注視領域を反転させた inverse mask-attention

$M_{\text{inverse}}(\cdot)$ は以下のように算出する.

$$M_{\text{inverse}}(\mathbf{s}_t) = 1 - M(\mathbf{s}_t), \quad (4.8)$$

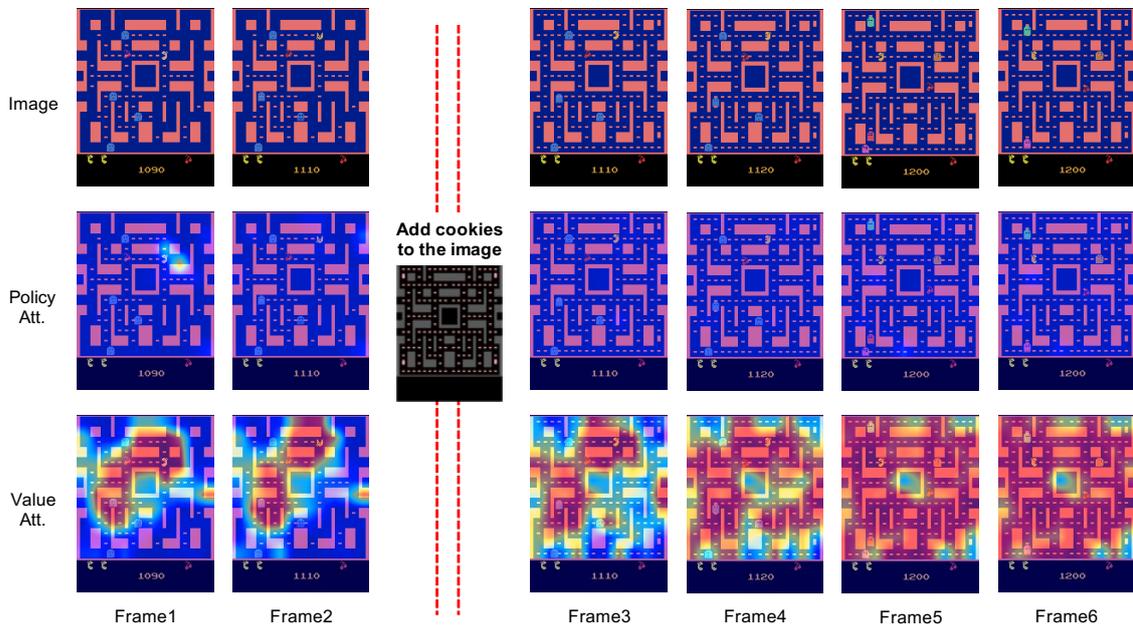
ここで, $bf s_t$ は状態 (ゲーム画面のグレースケール画像) である.

表 4.4 は, mask-attention 反転ありとなしのゲームスコア比較を示し, 図 4.6 は, mask-attention 反転によるゲームスコアの減少率を示している. 表 4.4 に示すように, 全てのゲームにおいて, mask-attention を反転するとスコアが大幅に低下することが確認できる. 特に BO と BR では, Mask-attention Loss を用いたエージェントモデルにおいて, mask-attention 反転後のスコアは random (行動のランダム選択) よりも低いことが分かる. MP, SQ, SI, FD では, 図 4.6 に示すように, 平均スコアが 85% 以上減少しており, 他のゲームと同様の傾向が確認できる. したがって, mask-attention の反転によって大幅なスコア減少が確認できた. この結果から, policy branch における mask-attention の注視領域が, 高いゲームスコアを達成するエージェントの行動選択に寄与している領域を示していることを確認した. また, Mask A3C と Mask A3C MaskattL の減少率が近いことから, Mask-attention Loss で制限した注視領域はエージェントの意思決定プロセスを分析する上で有効な領域を示していると考えられる. このように, Mask-attention Loss で注視領域を制限することで, エージェントの意思決定プロセスをより明確に示す解釈性の高い mask-attention を獲得可能であることを示した.

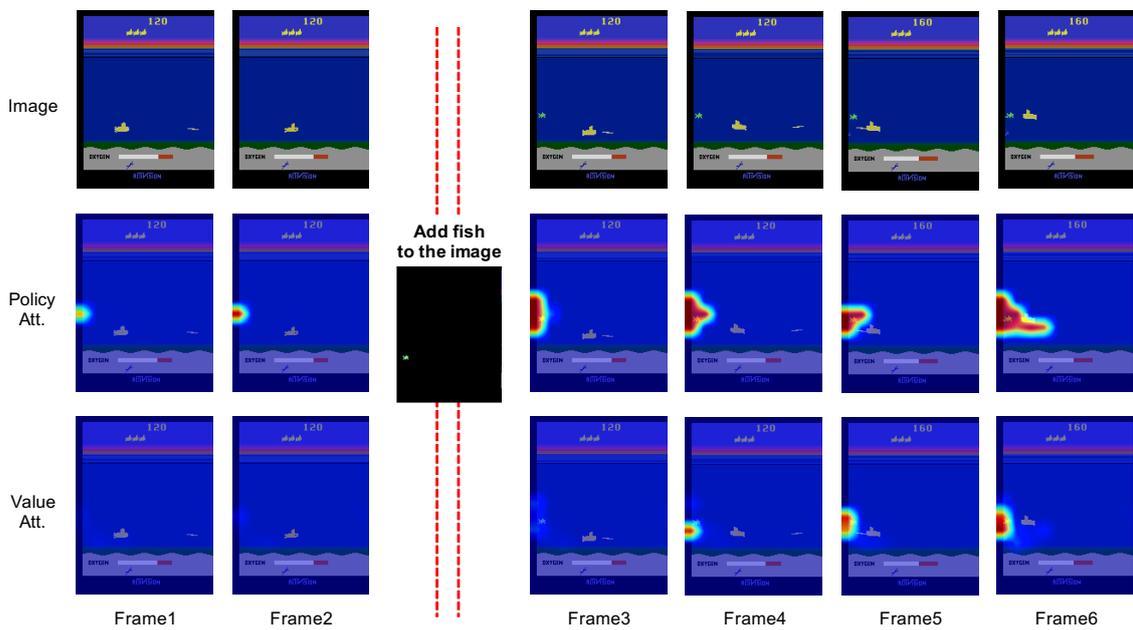
4.2.7 mask-attention に着目した新しい状態に対するエージェントの反応

4.2.4 節で述べたエージェントの注視対象に, 予期せぬ変化があった場合の mask-attention とエージェントの行動選択への影響を調査した. 注視対象の変化がエージェントの行動選択や mask-attention に大きく影響する場合は, その対象がエージェントの意思決定にとって重要な要素であると考えられ, mask-attention で示されたエージェントの注視対象が正しいことを意味する. 本実験では, mask-attention で示されたエージェントの注視対象として, Ms. Pac-Man (MP) におけるクッキーと, Seaquest (SQ) における魚 (SQ) を用いた. 本実験に用いるエージェントモデルは, 4.2.4 節で mask-attention 可視化に用いた Mask-attention Loss を用いた Mask A3C MaskattL とした. 調査方法として, エージェントモデルの評価時に想定外のフレームでクッキー (MP の場合), または魚 (SQ の場合) をエージェントモデルへの入力画像に追加する. MP では, エージェントが画面上に存在するクッキーの半分を取得し終えたタイミングで, 獲得したはずのクッキーを追加した. SQ では, 本来魚が存在しないフレームにおいて, 魚を追加した. それぞれオブジェクトを追加後, エージェントの行動選択と mask-attention がどのように変化するか調査する.

図 4.7a は, MP におけるクッキー追加前後に対するエージェントの行動選択と mask-attention の変化を示している. value mask-attention から, クッキー追加前のフレームでは, エージェントは画面に残っているクッキーを注視している. それに対して, クッキー追加後の Frame 2 以降では, エージェントは追加されたクッキーも含めた全てのクッキーに対し注視していることが確認できる. これら



(a) Ms. Pac-Man におけるクッキーに対するエージェントの反応.



(b) Sequest における魚に対するエージェントの反応.

図 4.7: mask-attention による新しい状態へのエージェントの反応の可視化. これは Mask A3C MaskAttL を用いた mask-attention の可視化である.

の結果から、MPのクッキーはエージェントの行動選択に大きく寄与するオブジェクトであると考えられる。

図4.7bは、SQにおける魚を追加前後に対するエージェントの行動選択とmask-attentionの変化を示している。図4.7b示す通り、Frame1と2におけるpolicy mask-attentionは、エージェントが魚が現れた画面左を注視していることに対し、Frame2以降のpolicy mask-attentionは、エージェントが追加した魚を強く注視し続けていることが確認できる。また、Frame1と2のvalue mask-attentionは注視領域がなく、Frame2以降は、エージェントが追加した魚に強く注視している。これらの結果から、SQにおいて魚がエージェントの行動選択に大きく貢献するオブジェクトであることを示している。

SQにおけるpolicy mask-attentionは、魚を追加直後のFrame3から魚に対し強く注視していることが分かる。つまり、policy mask-attentionは、魚が追加直後にすぐに影響を受けることを意味する。一方で、SQにおけるvalue mask-attentionは、Frame3において魚を強く注視せず、その後徐々に注視していることが分かる。同様に、MPにおけるvalue mask-attentionは、Frame3(クッキー追加直後のフレーム)で即座に全てのクッキーを注視するのではなく、徐々にクッキー全体を注視する傾向が確認できる。これらの結果から、注視対象に変化が生じた際のpolicy mask-attentionとvalue mask-attentionの影響が異なることが分かる。つまり、方策と状態価値の2つの観点に対するこれらのmask-attentionは、エージェントの意思決定に関する異なる知識を示していると言える。注視対象の変化に対して、policy mask-attentionの効果は即時的であったが、value mask-attentionの効果は数フレーム遅れて現れた。したがって、policy mask-attentionはエージェントの現在の行動に寄与する直接的な領域を示し、value mask-attentionは時系列情報を考慮したゲーム特性に関連する間接的な領域を示すと考えられる。これらのことからMask A3Cは、これら2つの観点からエージェントの意思決定に対する判断根拠を示す高い解釈性を持つ手法だと言える。

4.2.8 ロボットマニピュレーションタスクへの応用

本節では、mask-attentionの有効性を評価するために、ロボットマニピュレーションタスクによる評価実験について述べる。

■ ロボットマニピュレーションタスクの実験詳細

図4.8は、ロボットマニピュレーションタスクとエージェントモデルの概要を示している。本実験では、NVIDIA社が提供している強化学習研究用に開発した物理シミュレーション環境であるIsaac-gym [74]を用いたロボットマニピュレーションタスクを対象とする。このタスクは、トレイ内に存在する複数物体の中から、対象物体を多関節単腕アームロボットであるFranka Emika社のPanda¹を用いて把持するものである。本タスクで使用する物体は、ARC2017 RGBD Dataset [146]に含まれるダンベル、ラップ、テニスボールケースであり、把持対象とする物体はダンベルである。これらの物体は、

¹<https://franka.de/>

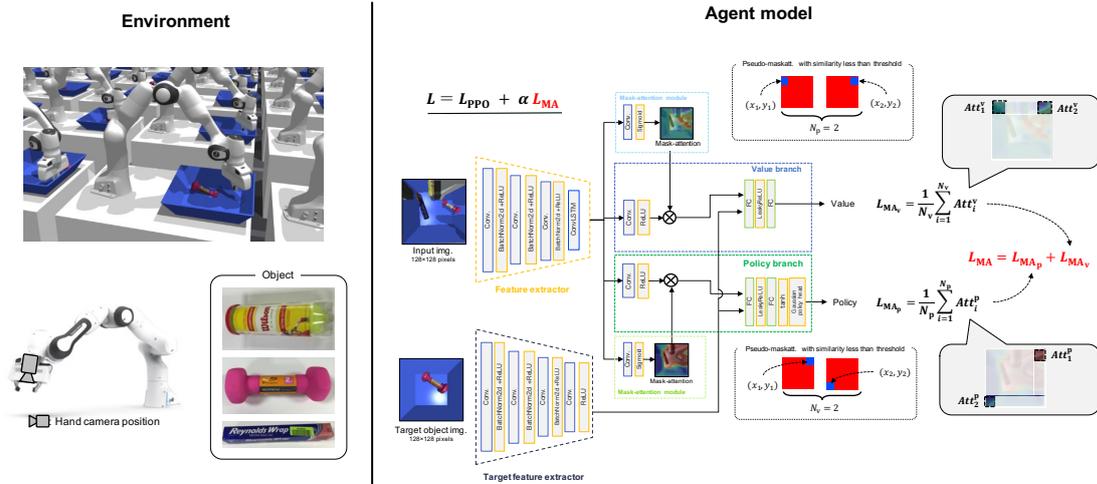


図 4.8: ロボットマニピュレーションにおける Mask A3C の概要.

表 4.5: ロボットマニピュレーションタスクにおける実験条件.

比較手法	PPO, Mask PPO, Mask PPO MaskAttL	
エージェントモデルの学習条件	optimizer	Adam
	training steps	2.5×10^7
	learning rate	0.0002
	discount rate	0.99
	termination condition of an episode	12 step
	threshold value for select pseudo-maskatt	0.01
	start step of Mask-attention Loss	2.0×10^7

エピソードごとでトレイ内にランダムな位置に配置される。エージェントの制御対象は Panda のエンドエフェクタであり、エンドエフェクタ前方に RGB カメラを設置し、ハンド視点画像がエージェントの入力情報である。本実験におけるエピソードの詳細を述べる。本実験では、エピソードの開始と同時にエンドエフェクタが自動降下し、12 ステップでトレイに到達する。12 ステップ後、エンドエフェクタは自動的に閉じられ、その後、エンドエフェクタの上昇が開始する。つまり、本タスクの目的は、12 ステップ以内に把持対象物体であるダンベルを把持できるようにエンドエフェクタを制御することである。エージェントモデルが選択できる行動は、トレイに対してエンドエフェクタを前後に移動させる、左右に移動させる、エンドエフェクタを左右回転させるという行動であり、これら行動は連続値である。これらの行動は、エンドエフェクタの下降中のみに行われる。報酬は、把持対象物体であるダンベルを一定以上の高さまで把持した場合のみに +1 である。エージェントモデルの学習には、ロボット制御タスクで高い性能を達成している Actor-Critic 法にもとづく深層強化学習アルゴリズムである Proximal Policy Optimization (PPO) [41] を用いる。

表 4.5 に本実験の詳細を示す。ここで、比較手法である Mask PPO (後述) は mask-attention module を導入した提案手法を指し、Mask PPO MaskattL は Mask-attention Loss を導入した Mask PPO を指す。学習条件にも示す通り、Mask PPO MaskattL の Mask-attention Loss は学習ステップ数が 2.0×10^7 ステップ以降に導入した。つまり、学習ステップが 2.0×10^7 ステップに達するまでは、PPO および Mask PPO と同じ損失関数である。学習後期から Mask-attention Loss を導入した理由は、ビデオゲームタスクの場合と同じである (4.2.1 節参照)。Mask-attention Loss の計算に用いる pseudo-maskatt を特定するための閾値は 0.01 と設定した。本実験では以下の 2 つの指標を用いて評価する。

- mask-attention を用いたエージェントモデルの視覚的説明
- ダンベルの把持成功率による評価

ロボットエージェントモデルの実装詳細。 ロボットマニピュレーションタスクでの実験に用いたエージェントモデルの詳細を述べる。エージェントへの入力情報は、エンドエフェクタに取り付けたカメラで撮影したハンド視点画像と、把持対象物体の画像であり、出力はエンドエフェクタの制御値である。入力されるハンド視点画像と把持対象物体の RGB 画像は、 128×128 サイズにリサイズする。Feature extractor は、ハンド視点画像から特徴量を抽出するモジュールであり、3 層の畳み込み層 (畳み込み処理後にバッチ正規化と ReLU を適用) と ConvLSTM から構成する。ここで、畳み込み層は出力次元が 16 の層、32 の層、64 の層から構成される。ConvLSTM の隠れ状態の出力次元は 64 である。Target feature extractor は、把持対象物体の画像から特徴抽出するモジュールであり、feature extractor と同じ構造 (ただし ConvLSTM を除く) に加え、出力次元 32 の畳み込み層を導入している (畳み込み処理後に ReLU を適用)。Value branch は、出力次元 32 の畳み込み層 (畳み込み処理後に ReLU が適用される)、全結合層が 2 層、および LeakyReLU [147] (全結合層間で適用) から構成する。Policy branch は、hyperbolic tangent 関数 (tanh) と、tanh を適用した値を平均値としたガウス分布を生成する Gaussian policy head に加えて、value branch と同じモジュールから構成する。tanh 前の全結合層は、エージェントの行動数と同じ出力ユニット数を持つ。各 branch において、エージェントは、target feature extractor により抽出した特徴ベクトルと、それぞれの branch における全結合層前の入力値と concat することにより、把持対象物体を考慮した把持動作を学習を実現する。本実験における PPO は、mask-attention module を除いた上述の構造と同じである。

■ mask-attention によるエージェントモデルの視覚的説明

本節では、ロボットマニピュレーションタスクにおける mask-attention の有効性の確認を目的とした評価実験について述べる。本実験では、4.2.8 節で述べたロボットマニピュレーションタスクにおいて、エージェントの意思決定プロセスを mask-attention の可視化から解析する。比較として、ビデオゲームタスクの評価実験 (4.2.3 節参照) でも用いた Greydunus らの手法 [4] による顕著性マップも可視化する。

mask-attention の可視化例を図 4.9 に示す。ここで、mask-attention とは Mask PPO MaskattL モデル (Ours) を用いて可視化されたエージェントの注視領域を示すヒートマップのことである。Scene 1

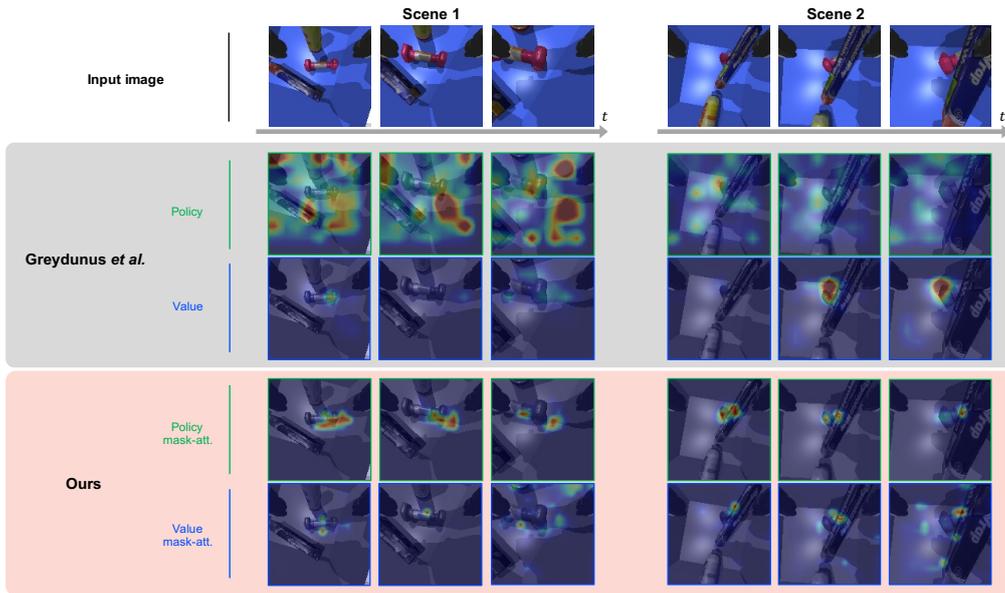


図 4.9: ロボットマニピュレーションタスクにおける mask-attention の可視化例. Ours は Mask PPO MaskattL による mask-attention の可視化である.

は把持対象物体であるダンベル全体が画像に映りこんでいるシーン, Scene 2 はダンベルがラップによって部分的に隠れてしまっているシーンである. Scene 1 における ours の policy mask-attention から, エージェントは一貫してダンベルを注視し, 他の物体に着目していないことが確認できる. これは, エージェントがダンベルを把持すべき物体として正しく認識し, ダンベル以外の物体を注視しないことで, ダンベルのみを把持するように制御していることを意味する. Scene 1 における ours の value mask-attention も, 同様にダンベルを注視していることが確認できる. また value mask-attention では, エージェントはダンベルの持ち手部分を強く注視している. これは, エージェントがダンベルを把持するための重要な部位を認識していることを意味する. 上述した Scene 1 におけるこれら mask-attention の傾向は, オクルージョンが発生する Scene 2 において同様に確認できる.

Greydunus らの手法と比較すると, policy の可視化例では Greydunus らの手法はノイズと思われる注視領域が多く見られ, 解釈が困難なマップとなっている. 一方で, ours は正確に把持対象物体を捉えていることが確認できる. また, Value の可視化例では, Greydunus らの手法と Ours の両方が把持対象物体を注視しているが, 上述の通り, ours は把持対象物体のより局所的な領域を捉えている. したがって, mask-attention は, Greydunus らの手法による顕著性マップより, エージェントの注視領域を明確に示し, 高い解釈性も保有していると言える.

これらの結果から, エージェントはオクルージョンが発生していても複数の物体から把持対象を正しく認識し, 把持する上で重要な部位 (ダンベルの持ち手など) を理解し, エンドエフェクタを制御していると考えられる.

表 4.6: ロボットマニピュレーションタスクにおける 1,000 エピソード間の把持成功率

Mask-attention module	Mask-attention Loss	grasping rate [%]
		54.58
✓		55.91
✓	✓	56.18

■ ダンベルの把持成功率による評価

本節では、ロボットマニピュレーションタスクにおけるエージェントの制御性能に対する mask-attention module と Mask-attention Loss の効果について述べる。本実験では、把持対象のダンベルに対する把持成功率によって、mask-attention module と Mask-attention Loss の効果を確認する。mask-attention module と Mask-attention Loss の有無による把持成功率を表 4.6 に示す。把持成功率は、学習済みモデルを用いた評価に対する 1,000 エピソード間の平均値である。mask-attention module なし (PPO) の把持成功率 54.58 に対し、mask-attention module あり (Mask PPO) の把持成功率は 55.91 であり、1.33 pt の向上していることが確認できる。また Mask PPO の把持成功率 55.91 に対し、Mask-attention Loss を導入した Mask PPO の把持成功率は 56.18 であり、0.27 pt の改善が確認できる。これらの結果から、mask-attention module と Mask-attention Loss の導入により、最適なロボット制御のために重要な領域をより明確に認識することで、ロボットの性能向上に貢献したと考えられる。

上述の mask-attention によるロボットエージェントモデルの視覚的説明と、ロボットマニピュレーションにおける把持成功率による評価から、本手法がビデオゲーム攻略のような 2 次元環境だけでなく、ロボット制御のような 3 次元環境においても有効であることを確認した。

4.2.9 ユーザによるエージェントモデルの行動予測にもとづく mask-attention の解釈性評価

本節では、mask-attention がユーザにとってエージェントの意思決定が理解可能なマップであるかどうかを調査することで、mask-attention の解釈性を評価する。

■ 調査方法

本実験では、エージェントの振る舞いについて、ユーザへ mask-attention を提示した場合に、mask-attention からユーザがエージェントの行動を予測できるかを検証する。つまり、ユーザが提示された mask-attention からエージェントの行動を予測できる場合は、その mask-attention はユーザにとって解釈性の高いものであると言える。また比較として、注視領域を提示しない教示方法と、Greydanus らの手法に基づく顕著性マップを用いた教示方法も同時に検証する。ここで、注視領域を提示しな

い教示法は、エージェントへの入力情報である RGB 画像のみを提示することで用教示する。本実験では、51 人の被験者を教示方法を軸に 3 グループに分け、グループごとに 2 種類の設問に回答してもらう。1 つ目の設問は、エージェントの行動を予測する設問である。この設問は、グループごとの教示方法で提示されたマップから、与えられたフレームにおけるエージェントの行動を予測し、選択肢形式で回答してもらうものである。回答の選択肢は、そのタスクでエージェントが取り得る行動であり、unknown を追加する。もし、被験者が与えられた情報からエージェントの行動を判断できない場合は、unknown を選択してもらう。2 つ目の設問は、把持対象物体の予測に関する設問である。この設問は、エージェントが把持しようとしている物体が 3 つの物体（ダンベル、ラップ、テニスボールケース）のうち、どの物体であるかを予測し、選択肢形式で回答するものである。これは、ロボットマニピュレーションタスクに関する設問であり、ロボットマニピュレーションにおけるエージェントの行動予測問題に回答後、本設問に取り組んでもらう。調査プロセスは以下の通りである。

1. 51 名の被験者を、教示方法ごとに 3 グループ（1 グループ 17 名）にグループ分けをする。各グループの概要は以下の通りである。
 - w/o attention：エージェントの注視領域に対する提示はなく、エージェントモデルの入力である RGB 画像を用いた教示方法。
 - Greydanus *et al.*：Greydanus らの手法による顕著性マップを用いた教示方法。
 - Ours：Mask-attention Loss を導入した本手法による mask-attention を用いた教示方法。
2. 被験者は、各タスクの説明動画を視聴し、エージェントが取り組むタスクが何であるかの理解を促す。このときの説明動画はタスクに対する理解を目的とするため、ランダム制御のエージェントによるデモンストレーション動画を使用する。つまり、学習済みエージェントによるデモンストレーションは使用しない。
3. 被験者は、グループごとの教示方法で提供されたマップをもとに、エージェントの行動予測問題（各タスク 10 問、全タスク 30 問）に選択肢形式で回答する。
4. 被験者は、2 と 3 の手順を全タスク（ビデオゲーム攻略:3 タスク、ロボットマニピュレーション:1 タスク）において取り組む。ロボットマニピュレーションタスクの全設問に回答したのち、被験者は把持対象物体に関する予測設問に回答する。
5. 各グループの被験者によるエージェントの行動予測設問に対する正答率と、把持物体物体の予測設問に対する正答率から、各教示方法の解釈性を分析する。

これらの設問に対する正答率が高い場合は、ユーザがその教示方法によって提示されたマップから、エージェントの行動を理解できることを示している。言い換えれば、そのマップはユーザーにとって解釈性が高いということの意味する。

表 4.7: エージェントの行動予測に対する設問の平均正答率と unknown 回答率. w/o attention は RGB 画像のみの提示による教示方法である.

ユーザへの教示方法	平均正答率 [%]			unknown 回答率 [%]
	ビデオゲーム攻略	ロボット制御	全タスク	
w/o attention	49.60	22.94	42.94	3.67
Greydanus <i>et al.</i> [4]	32.74	12.94	27.79	8.67
Ours	73.33	44.11	66.02	2.50

■ 調査結果

2種類の設問に対する正答率から、各教示方法に用いたマップの解釈性を分析する. 表 4.7 は、エージェントの行動予測の設問に対する平均正答率と unknown の回答割合を示している. 平均正答率が高いということは、教示に用いたマップがエージェントの行動を正確に表したマップであることを示し、unknown の回答率が低いということは、ユーザがエージェントの行動を読み取りやすいマップであることを示す. したがって、2つの評価指標でより良い評価値を得たエージェントの視覚的説明方法は、エージェントの行動に対する解釈性が高いと言える.

表 4.7 から、ビデオゲーム攻略とロボットマニピュレーションの正答率を比較すると、ロボットマニピュレーションの正答率はどの教示方法でも一貫して低いことが確認できる. ロボットマニピュレーションはビデオゲームよりも奥行き情報を考慮する必要がある3次元環境であり、エージェントの行動空間も広大である. そのため、提示されたマップとユーザの直感的な理解の結びつきが弱く、ロボットマニピュレーションの正答率はビデオゲームよりも低かったと考えられる. Ours の平均正答率は 66.02% で全タスクの中で最も高く、unknown の回答率は 2.5% と最も低いことが確認できる. これらの結果から、mask-attention (Ours) はエージェントの行動に対するユーザの理解を高め、エージェントモデルに対する解釈性の高い視覚的説明手法であると言える. 一方、Greydanus らの手法は w/o attention に比べて、全タスクにおいて 15.15% 正答率が低く、unknown の回答率も 5.0% 高いことが確認できる. この結果から、Greydanus らの手法は mask-attention よりも解釈性が低く、ユーザに誤った情報を提供してしまっていると考えられる.

図 4.10 に把持対象物体に関する予測設問に対する回答の内訳を示す. ここで、全ての教示手法において、エージェントモデルの把持対象はダンベル (dumbbells) である. つまり、この設問ではダンベルが正解となる. w/o attention では、47.1% の被験者がダンベルと回答したが、29.4% の被験者が unknown と回答した. この結果から、RGB 画像のみを用いた w/o attention では、把持対象がどの物体であるか理解するには不十分であったと考えられる. また Greydanus らの手法では、41.2% の被験者がダンベルと回答し、ラップとテニスボールケースと回答した被験者は 20 - 30% とばらつきが見られた. これに対し、Ours では 94.7% の被験者がダンベルと回答し、高い正答率であることが確認できる. したがって、Ours における mask-attention は Mask-attention Loss による注視領域の制約に起因して、エージェントが把持対象を理解する上で高い解釈性を有していると言える.

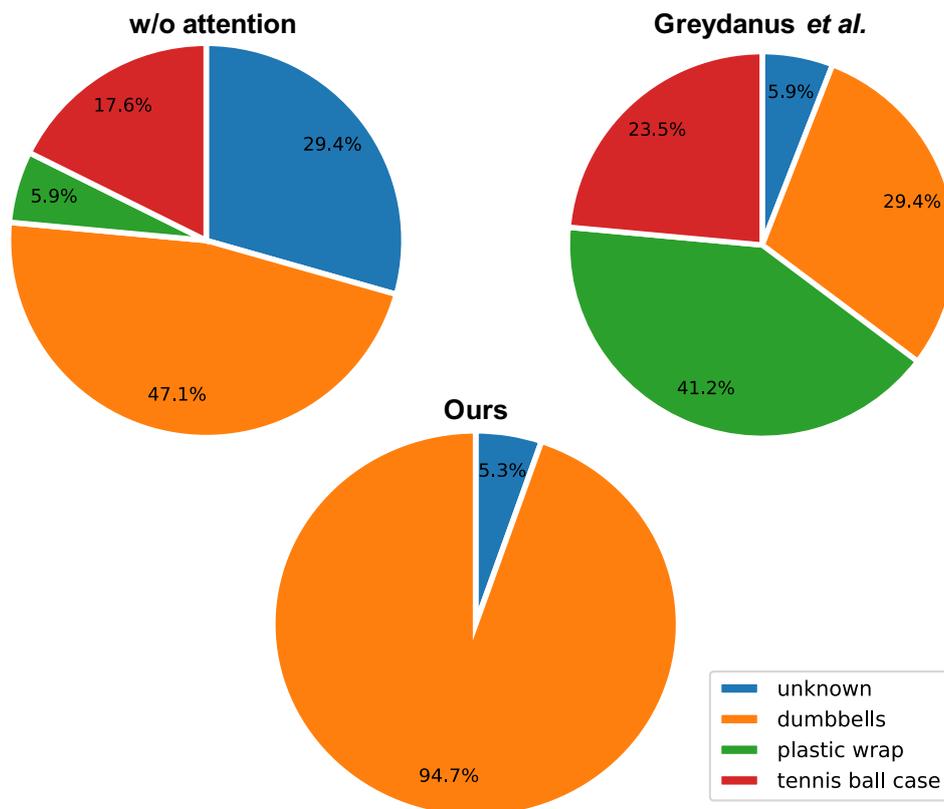


図 4.10: 把持対象物体の予測に対する設問の回答詳細

これらの結果から，mask-attention module と Mask-attention Loss が mask-attention の解釈性を高め，エージェントモデルの振る舞いを明確に示していることを確認した。

4.3 まとめ

本章では，深層強化学習の枠組みにおいて，エージェントモデルの振る舞いに関する判断根拠を明らかにするために，方策と状態値の2つの観点に着目し，Actor-Critic 法にもとづく深層強化学習エージェントの視覚的説明手法である Mask Attention A3C (Mask A3C) を提案した。Mask A3C は，Actor-Critic 法の output branch である方策と状態価値に対する attention 機構を導入し，各 branch の出力に関連する重要な領域を強調する mask-attention を生成する。深層強化学習において，方策は行動選択の確率分布を表し，状態価値は現状態に対する価値を表すため，これらは深層強化学習エージェントの意思決定において重要な要素である。そのため，これらの mask-attention を可視化することで，エージェントモデルの意思決定プロセスに対する網羅的な洞察を生成可能な視覚的説明手法を確立した。また同時に，エージェントモデルの学習過程で Mask-attention Loss を適用することで，エージェントモデルの意思決定に影響を与えない領域への注視を制限し，mask-attention の解釈性を向上させた。ビデオゲームとロボットマニピュレーションの実験結果から，mask-attention はエージェ

ントモデルの意思決定プロセスを解釈するための重要な領域を示していることを確認した。

本研究では、方策と状態価値の観点から actor-critic ベースな深層強化学習手法に着目したが、mask-attention module などはシンプルな構造であるため、value ベースな深層強化学習手法など、他の深層強化学習法への応用も可能である。Mask-attention Loss における pseudo-maskatt を特定する閾値と導入ステップは、タスクに最適な値に設定する必要がある。本研究の課題としては、これらのハイパーパラメータについての詳細な調査が挙げられる。また、深層強化学習エージェントの意思決定に焦点を当てた説明/解釈性に関する研究の多くは、本研究も含め、深層強化学習エージェントの意思決定に対する視覚的説明である。そのため、これらの研究によって提供される深層強化学習エージェントの attention map は、ユーザによって解釈が異なってしまうという課題に直面する。そこで、エージェントモデルの意思決定を自然言語で説明する手法を開発も課題として挙げられる。

第5章

選択可能な行動に着目した深層強化学習エージェントに対する視覚的説明

本章は、4章と同様に、説明性/解釈性の高い深層強化学習エージェントの実現を目指した説明可能な強化学習 (eXplainable Reinforcement Learning; XRL) について議論する。4章では、深層強化学習における方策と状態価値という2つの観点に焦点を当て、エージェントモデルの意思決定プロセスに対する網羅的な洞察を生成した。この手法は、Actor-Critic ベースな深層強化学習手法にもとづいているため、方策は確率分布で表現され、エージェントモデルが最適と判断した行動に限定して、エージェントモデルの視覚的説明を提供した。これは、エージェントモデルの振る舞いに焦点を当てた他の XRL 手法も同様であり、エージェントが選択した行動のみに対する XRL 手法が報告されている [4, 5, 148]。しかし、深層強化学習エージェントモデルの意思決定プロセスについてのさらなる網羅的な解析には、エージェントモデルが選択した行動だけに限定せず、エージェントモデルが選択可能な行動全てに対して洞察を提供することが有効であると考えられる。また XRL の文脈では、これらの視覚的説明情報をユーザに対してどのような形式で提供するかも重要な要素の一つである。例えば、実環境上でユーザとロボットがリアルタイムにインタラクションするようなロボット制御タスクでは、ユーザがロボットの動作を直感的かつ即座に理解できる枠組みが求められている。このようなロボティクス分野では、物理的特性に制約されることなく視覚フィードバックを提供可能な Augmented Reality (AR) が有効であると報告されている [149, 150, 151]。以上の背景を踏まえ、本章では、深層強化学習エージェントの意思決定プロセスに対する高い説明性/解釈性を目的とし、エージェントモデルが選択可能な行動全てに焦点を当て、行動情報を用いた transformer encoder-decoder 構造を導入した XRL 手法 Action Q-Transformer (AQT) を提案する。また、ロボット制御タスクのような実空間上における視覚的説明情報のユーザに対する効果的な提供を目的とし、物理的特性に制約されることなく可視化できる Augmented Reality (AR) に焦点を当て、エージェントモデルの意思決定プロセスに対する AR を用いた視覚的フィードバック手法を提案する。

本章の構成は次の通りである。5.1 節では、提案手法である深層強化学習エージェントである XRL 手法 Action Q-Transformer について述べる。5.2 節では、AR に着目し、エージェントモデルの意思決定プロセスに対する AR を用いたユーザへの視覚的フィードバック手法について述べる。5.3 節では、ビデオゲームタスクとロボットの自立移動制御タスクにおける評価実験について述べる。最後に、5.4 節で本章をまとめる。

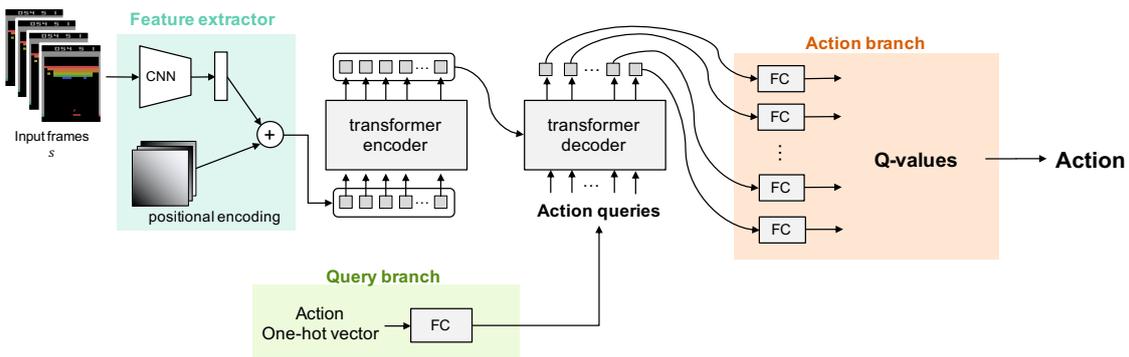


図 5.1: Action Q-Transformer 構造の概要

5.1 Action Q-Transformer

深層強化学習エージェントは、数多くの研究によって高い制御性能を獲得可能であると報告され、様々な制御タスクへの応用が期待されている。しかし、学習により獲得したエージェントモデルは、モデルが選択する行動に対して判断根拠が不明確であり、深層強化学習エージェントの信頼性低下を招き、実利用の障害となっている。そこで、Q-learning に基づく深層強化学習手法に Transformer encoder-decoder 構造を導入した XRL 手法 Action Q-Transformer (AQT) を提案する。

AQT は図 5.1 に示すように Feature extractor, Transformer Encoder-Decoder, Query branch, Action branch から構成される。Encoder では入力画像パッチ間の関係を考慮し、Decoder では Encoder の出力と行動情報を表す Action query との関係を考える。Action query は、エージェントが取りうる各行動情報を示し、行動ごとに対応した query を用いて Action branch で行動価値 (Q 値) を計算する。計算された Q 値が最も大きい行動が、その時刻におけるエージェントの行動として選択される。

5.1.1 Feature extractor

Feature extractor は、CNN を用いて入力画像から特定サイズの特徴マップを取得する。この取得した特徴マップは、位置情報を付加し 1 次元特徴量に変換する。1 次元特徴量の各要素は、特徴マップの各要素に対応する。つまり、特徴マップの各要素は、入力画像を特徴マップと同じサイズになるようにパッチ分割した際の 1 パッチに対応する。取得された 1 次元特徴量は transformer encoder の入力値として用いられる。

5.1.2 Transformer Encoder-Decoder

本手法では図 5.2 に示す transformer encoder-decoder を導入することで、エージェントの行動に対して解釈性/説明性の高い深層強化学習エージェントモデルを実現している。

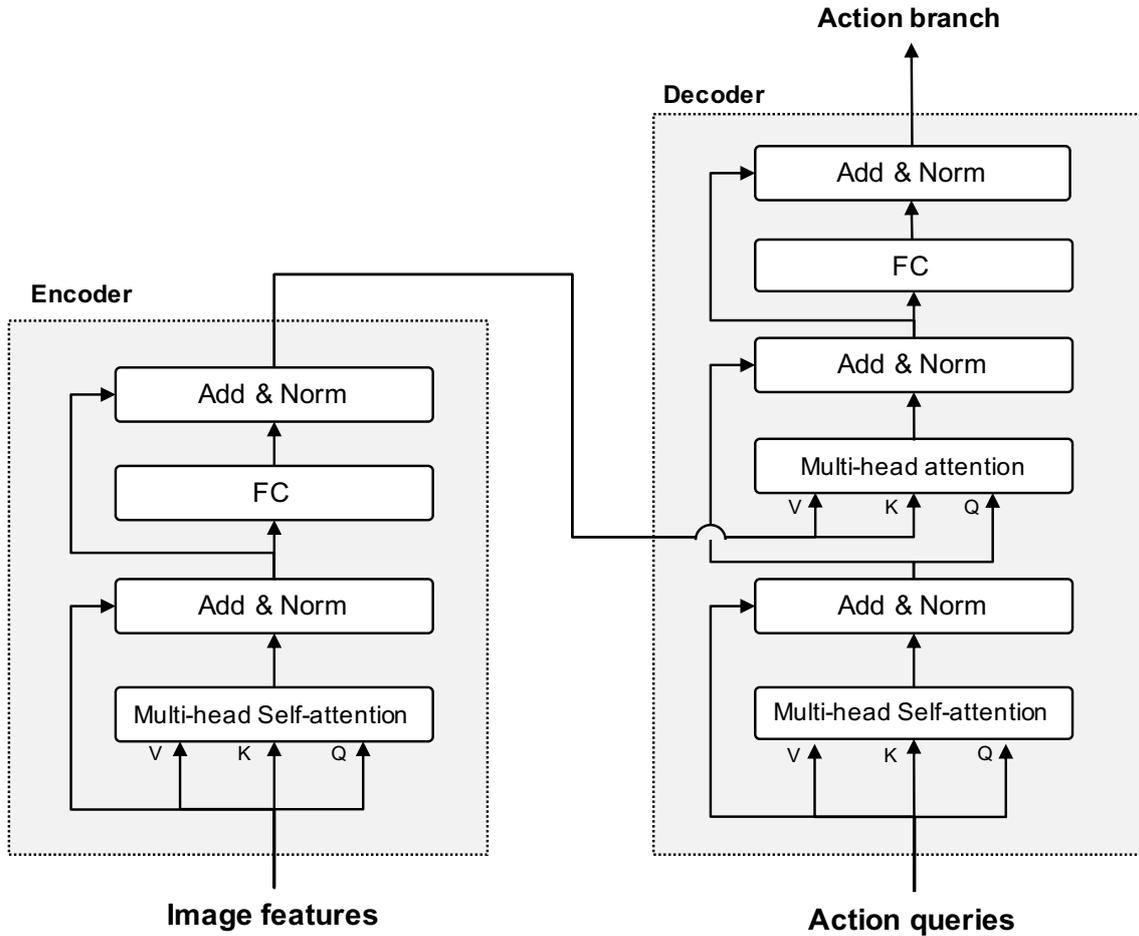


図 5.2: Transformer Encoder-Decoder の詳細

Encoder-Decoder structure. Encoder は Multi-head Self-attention, 全結合層 (FC), 残差機構, 正規化処理 (Add&Norm) から構成される. Multi-head Self-attention は, 入力である特徴マップにおける要素間の関係を self-attention によって考慮し特徴量を計算する. Decoder は, Encoder と同じモジュールに Multi-head attention を加えた 4つのモジュールから構成される. Multi-head attention の Value と Key は Encoder が取得した特徴量であり, Query は Query branch (後述) により算出した action query である. Decoder における Multi-head attention の attention weight は以下のように計算する.

$$att_{de}(Q_{act}, F_{en}) = \text{softmax} \left(\frac{Q_{act} \cdot F_{en}^T}{\sqrt{dim}} \right) F_{en}, \quad (5.1)$$

ここで, Q_{act} は multi-head attention 後の action query, F_{en} は encoder の出力値, dim は Q_{act} と F_{en} の次元数, $\text{softmax}(\cdot)$ は softmax 関数である. Decoder の出力を Action branch への入力とすることで, Decoder から行動ごとに固有の attention weight を獲得する. この action query を利用する Transformer encoder-decoder 構造を採用することで, エージェントの意思決定を理解する上で重要な

Query branch (Number of actions: 4)

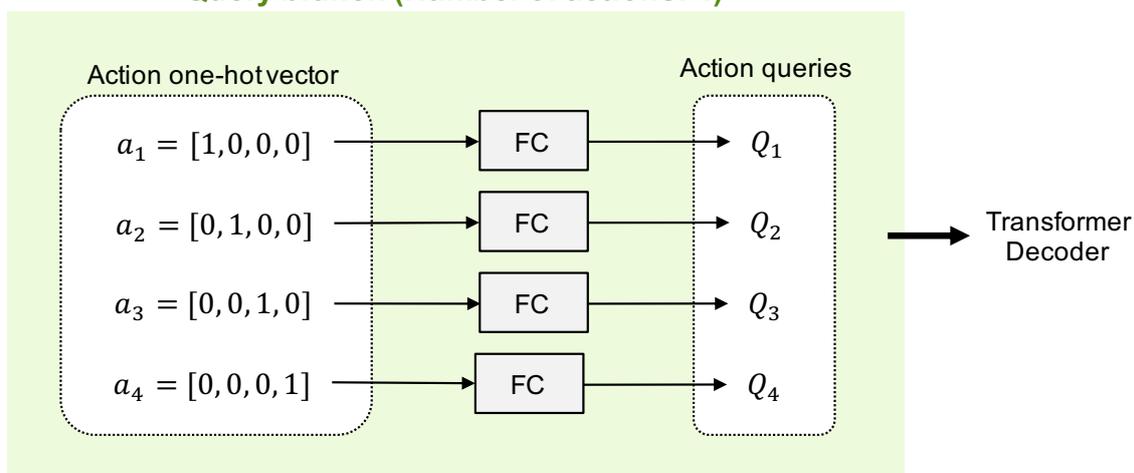


図 5.3: Query branch の詳細

要素となる行動に対する attention weight を獲得することができ、エージェントの意思決定に対して解釈性の高いエージェントモデルを実現する。

Query branch. 本手法では、各行動に対する Q 値の計算は、行動情報を Transformer-decoder の Query として利用することで実現する。そのため、Query branch で decoder に使用する action query を計算する。エージェントの行動数が 4 の場合における Query branch を図 5.3 に示す。Query branch の処理を以下で述べる。本手法は Value ベースな深層強化学習アルゴリズムにもとづいた手法であるため、エージェントの行動が離散的であると仮定して、各行動の One-hot vector を定義する。各 One-hot vector を全結合層により埋め込みベクトルへ変換し、Decoder の Query (action query) として使用する。ここで、Query 数は行動数と同じであり、埋め込みベクトルへの変換に用いる全結合層のパラメータは他の層と同様に学習中に更新される。これにより、学習可能なパラメータである action query を用いて各行動に対する Q 値が計算可能となる。

5.1.3 Action branch

Decoder の出力は action query ごとに算出される。つまり、行動ごとに Query 特徴ベクトルを算出する。Action branch では、各行動特徴量からヘッドの役割も持つ全結合層によって Q 値を算出する。ここで、行動数分の Q 値が算出され、最も Q 値が高い行動をエージェントの行動として選択する。この Action branch は、採用する value ベースな深層強化学習アルゴリズムによって、適宜構造を変更するものとする。

5.1.4 Target Trained Q-Network

AQT では Transformer encoder-decoder 構造を採用しているため、学習において transformer [2] と同じ性質を持っていると考えられる。transformer の性質として、transformer の性能は学習に用いるデータ量に応じて精度が向上するスケーリング則を持つことが知られている [66, 68]。したがって、これらの性質と照らし合わせ AQT の性能は学習ステップ数に依存すると考えられる。そこで、学習済みベースラインモデルの出力を目標値とする損失関数 Target Trained Q-network (TTQ) を提案し導入することで、上記問題の解決を試みる。ここでベースラインモデルとは、AQT を構築する際にベースとした既存の value ベースな深層強化学習アルゴリズムのことである。

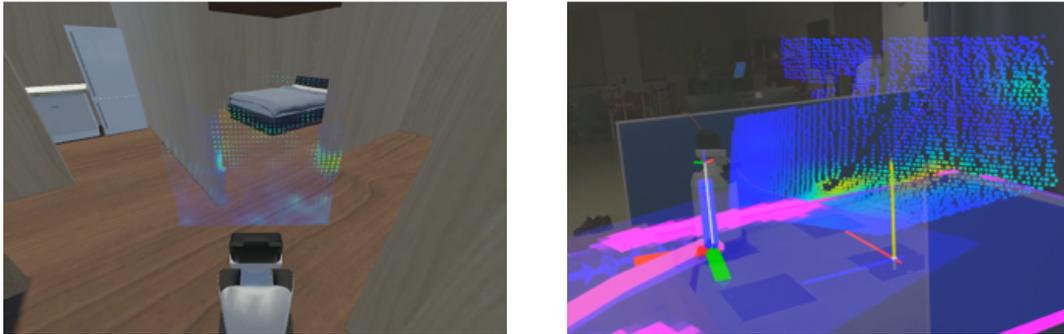
TTQ は、学習済みベースラインモデルが出力する行動価値 Q_{base} を目標値として、AQT モデルの出力 Q_{aqt} と目標値との二乗誤差として設計する。TTQ L_{ttq} を導入した AQT の損失関数 L は次式で計算される。

$$\begin{aligned} L &= L_{\text{aqt}} + \alpha L_{\text{ttq}}, \\ L_{\text{ttq}} &= (Q_{\text{base}}(s_t, a; \theta_{\text{base}}) - Q_{\text{aqt}}(s_t, a; \theta_{\text{aqt}}))^2, \end{aligned} \quad (5.2)$$

ここで、 L_{aqt} はベースラインとなる深層強化学習モデルと同様の損失関数、 α は L_{ttq} の学習率、 θ_{base} 、 θ_{aqt} は各モデルのネットワークパラメータである。

5.2 AR インターフェースを用いたユーザに対する視覚的フィードバック

エージェントモデルに着目した XRL の文脈では、エージェントモデルの意思決定プロセスに対する説明や解釈を、ユーザに対しどのように提供するかが非常に重要な要素の一つである。一般的な視覚的説明手法では、入力に対する顕著性マップや attention map を入力画像に重畳する形で、画像として可視化することが多い。しかし、実空間上で動作するロボット制御タスクなどでは、画像による視覚的フィードバック方法は奥行きなどの 3 次元空間情報が理解しにくく直感的ではないため、ユーザの理解には不向きである。そこで、Augmented Reality (AR; 拡張現実) インターフェースに着目する。AR は、現実世界に仮想的な視覚情報を重ね合わせる技術であり、物理的な制約なしに表現力豊かな視覚的フィードバックを提供することができる。この利点を用いて、深層強化学習により獲得した高性能なロボットエージェントの意思決定プロセスに対する AR を用いた視覚的フィードバック手法を提案する。可視化方法としては、ロボットの深度センサから得られる奥行き情報を点群に変換し、attention weight の値に応じて点に色付けを行う。ここで可視化に用いる attention weight は、Q 値が最も高い行動に対する AQT の decoder-attention である。AR による attention weight の可視化例を図 5.4 に示す。図 5.4 (a) は CG 環境における AR を用いた視覚的フィードバックのシミュレート例、図 5.4 (b) は実世界における AR デバイスによる視覚的フィードバック例である。ここで、可視化した attention weight はヒートマップで表現され、赤はロボットエージェントが強く注視、青は



(a) CG 環境：シミュレーション環境において AR を用いて attention weight を可視化。 (b) 実環境：実世界において AR を用いて attention weight を可視化。

図 5.4: AR を用いた attention weight の可視化例。

ロボットエージェントが注視していないことを表す。この可視化結果を用いることで、深層強化学習によって獲得したロボットエージェントの意思決定を、ユーザが直感的に理解しやすい形でフィードバックすることができる。

5.3 評価実験

本節では、AQT の有効性を評価するために、OpenAI gym [145] のビデオゲーム攻略タスク、および屋内環境におけるロボットの自律移動タスクを用いた評価実験について述べる。ロボットの自律移動タスクでは、Action Q-Transformer (AQT) の性能評価とともに、AR インタフェースを用いた視覚的フィードバックについての評価も述べる。以下で、それぞれの評価実験と結果について明記し考察する。

5.3.1 ビデオゲーム攻略タスク

本節では、OpenAI gym のビデオゲームタスクを用いた評価実験について述べる。この実験では、Atari 2600 の 50 ゲームを用い、attention weight を可視化によるエージェントの意思決定に対する視覚的分析には、Breakout と Seaquest を用いる。本実験における AQT のベースライン手法として、Atari 2600 にて高いスコアを獲得している深層強化学習アルゴリズムである Rainbow [1] を採用する。比較手法は、ベースラインモデルである Rainbow と本手法である AQT である。各手法の学習終了条件は、学習ステップ数が 5.0×10^7 に達したときである。またエピソードの終了条件は、ゲームの 1 プレイ終了か、ステップ数が 1.08×10^5 に達した時である。評価方法は以下の 2 通りである。

- Atari 2600 におけるスコア比較
- attention weight を用いたエージェントモデルの視覚的説明

■ ビデオゲームエージェントモデルの詳細

Rainbow は, dueling network [27] を採用していることから, Q 値算出にあたり状態価値と advantage に分けて出力する必要がある. したがって, 本実験における AQT モデルの Action branch は, 状態価値を算出する Value branch と, advantage 値を算出する Advantage branch の 2 つを構築する. Value branch では encoder の出力値を, Advantage branch では decoder の出力値を各 branch への入力値として用いる. これにより, encoder-attention は状態価値, decoder-attention は advantage に寄与する固有の attention weight を獲得する構造となる. Value branch と Advantage branch では, branch ごとに状態価値と advantage を計算し, 最終出力である行動価値を求める. 行動価値 $Q(s_t, a)$ は, 状態価値 $V(s_t)$ と Advantage $Adv(s_t, a)$ を用いて, 次のように計算される.

$$Q(s_t, a) = V(s_t) + Adv(s_t, a) - \frac{\sum_{i=1}^{n_a} Adv(s_t, a_i)}{n_a}, \quad (5.3)$$

ここで, a はエージェントの行動, s は環境の状態, t は時刻, n_a はエージェントの行動数を示す.

■ Atari2600 におけるスコア比較

Atari2600 での 100 エピソード間の平均スコアを図 5.5 に示す. Bowling や Breakout のようなゲームでは, AQT (灰色) はベースラインの Rainbow (赤破線) よりもスコアが高いことが確認できる. これらのゲームではエージェントの行動数が少なく, 環境の探索空間が小さい特性を持つタスクであるため, 比較的単純なビデオゲームタスクである. したがって, transformer 構造を用いた AQT は, 各行動の Q 値を正確に推定し, それによってスコアを向上したと考えられる. 一方, Fishing Derby や Seaquest では AQT (灰色) によるスコア向上が確認できない. つまり, AQT のスコア向上が確認できなかったゲームは, スコアが向上したゲームよりも学習が困難であったと言える. このような学習が困難なタスクでは, 5.1.4 節で述べた transformer の スケーリング則の影響が特に大きいと考えられる. したがって, Seaquest のような学習が難しいゲームタスクでは, AQT のスコアはベースラインの Rainbow のスコアよりも低いと考えられる.

実験における TTQ を導入した AQT (AQT+TTQ) は, AQT の損失関数 L_{aqt} による性能向上を考慮するため, TTQ の学習率 α を 1.0 から 0.0 まで 2.5×10^7 ステップで線形減衰させた. つまり, 2.5×10^7 ステップ後の学習は AQT と同じである. 図 5.5 から AQT+TTQ は, AQT が高い性能を獲得できなかったタスク (Beamrider, Fishing Derby, Seaquest など) において, Rainbow に比べてスコアの向上が確認できる. 一方, AQT でスコア向上が確認できたタスク (Bowling, Breakout など) では, TTQ を導入していない AQTの方がスコアが高かった. これは, 目標とする Rainbow のスコアが AQT のスコアより低かったため, TTQ が AQT の学習を妨げる結果となったと考えられる.

これらの結果から, 本手法 (AQT, AQT+TTQ) がベースラインの Rainbow を上回ることが実証し, Transformer encoder-decoder 構造の有効性を確認した.

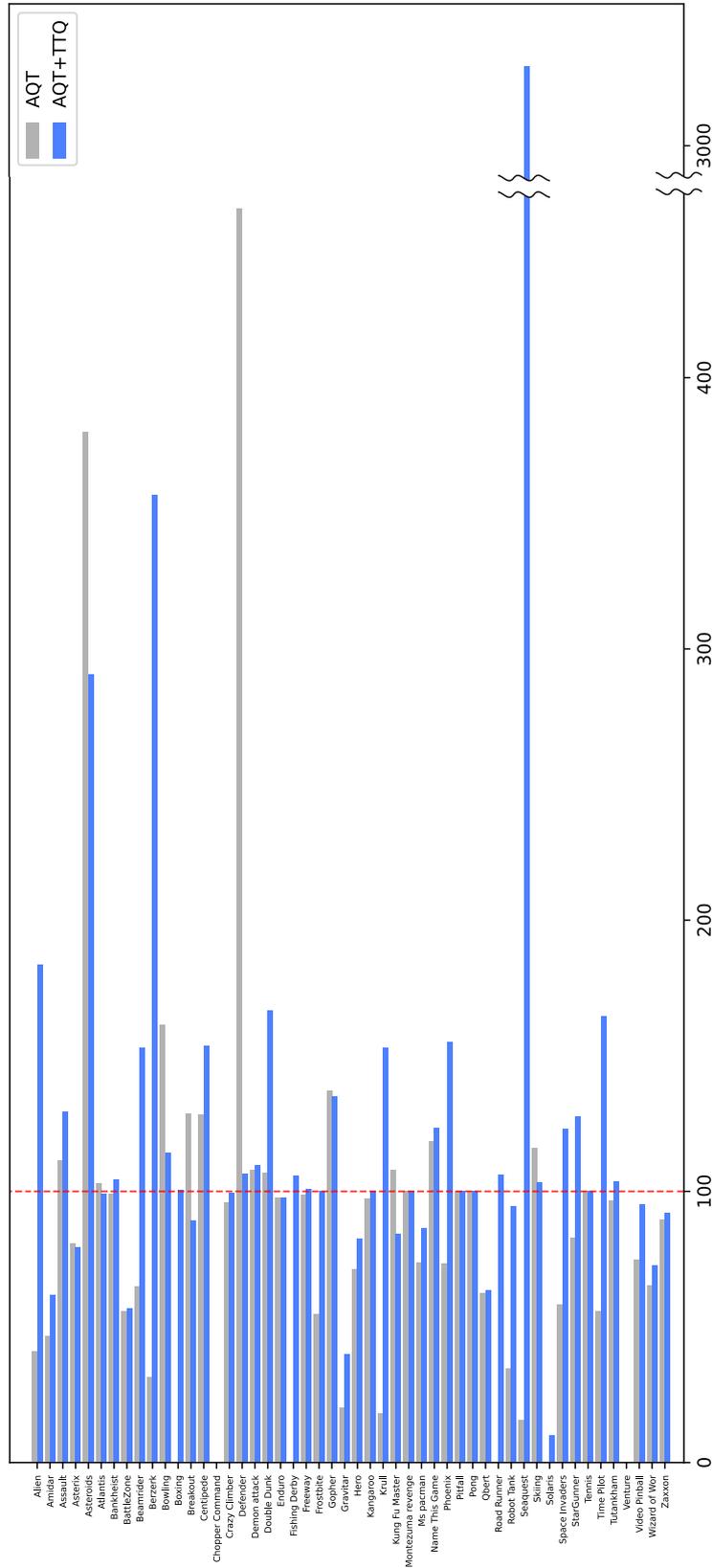
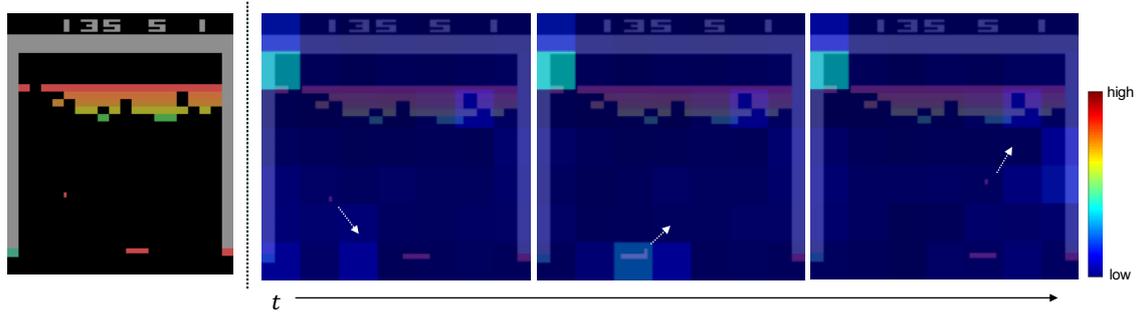
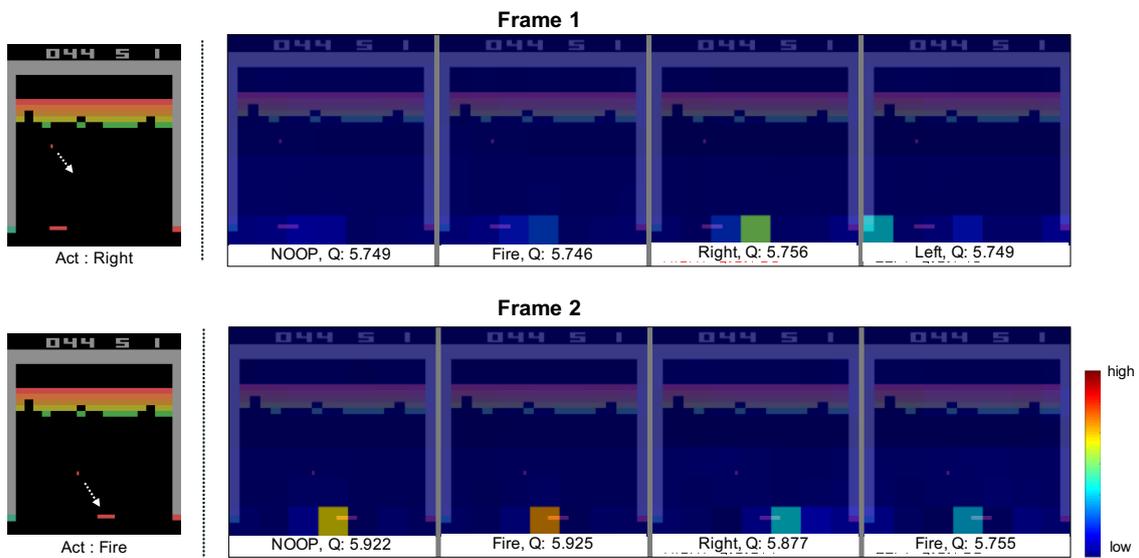


図 5.5: Atari 2600 における 100 エピソード間の平均スコア. グラフは、ベースラインである Rainbow モデル (赤破線) の平均スコアを 100 としたときの、各手法の平均スコアに対する割合を示している.



(a) Encoder-attention: パドルでボールを打ち返すシーンの可視化例。この可視化例は左から右に時刻方向に横並びとなっている。



(b) Decoder-attention: attention weight 下のキャプションは、行動の名前と現フレームにおける Q 値を示す。左に示す入力画像の下に記載されたキャプションは、現フレームでエージェントが選択した行動を示す。また Breakout における“Fire”は、“Noop”と同じ振る舞いをする。つまり、“Fire”と“Noop”はどちらも動作なしである。

図 5.6: Breakout における attention weight の可視化例。画像上の白矢印は、ボールの進行方向を示している。

■ attention weight を用いたエージェントモデルの視覚的説明

Breakout と Seaquest において、それぞれ図 5.6, 5.7 に示すように attention weight を可視化した。ここで、各ゲームでは最もスコアが高かったモデル（つまり、Breakout では AQT モデル、Seaquest では AQT+TTQ モデル）を用いて attention weight を可視化した。以下では、これらの図に示した attention weight からエージェントの意思決定について解析する。

Breakout. Breakout は、パドルでボールを打ち返して画面上部に位置するブロックを破壊するゲームである。エージェント (= パドル) の行動は次の 4 つである：Noop, Fire, Left, Right.

Encoder-attention (状態価値). ボールを打ち返すシーンにおける encoder-attention の可視化例を図 5.6a に示す。ここでエージェントは、ボールが打ち返される瞬間のパドルと、ブロック数が最も少な

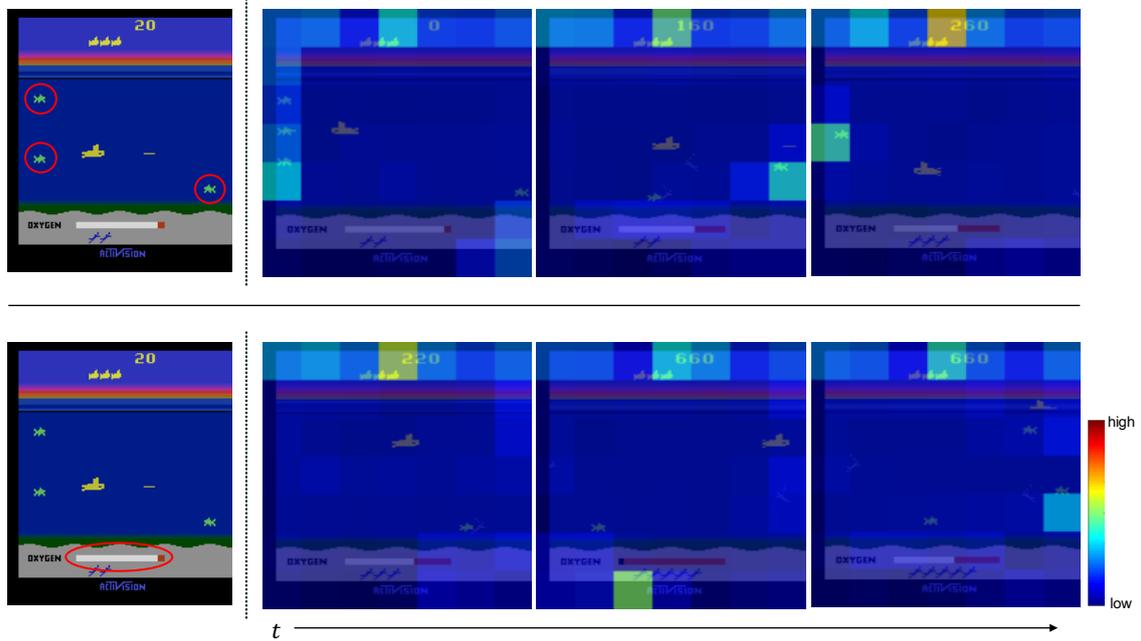
い領域を注視していることが確認できる。Breakout では、ブロックを破壊することでスコア獲得となる。したがって、エージェントにとって重要な状態とは、パドルでボールを打ち返した状態であると考えられる。また、エージェントがブロックに穴を開け、その穴に向かってボールを打ち返すと、ボールはブロック上部の空間でバウンドを繰り返し、多くのブロックを破壊することができる。以上のことから、エージェントはボールを打ち返す瞬間にパドルとボールが重要な物体であり、ブロックに穴を開けることが高得点を得るために重要な状態であることを正しく認識できていると考えられる。

Decoder-attention (advantage). Breakout における decoder-attention の可視化例を図 5.6b に示す。Frame 1 において、エージェントはパドルの右側を注視していることが確認できる。Breakout では、エージェントの Right はパドルを右に動かし、Left はパドルを左に動かす行動である。したがって、エージェントは操作対象がパドルであることを認識し、各行動によるパドルの移動先を予測していると考えられる。Frame 2 では、エージェントは全ての行動においてパドルを注視しており、行動価値が大きいほどエージェントは強くパドルを注視している。Frame 2 は、ボールがパドルに近づいているシーンのため、エージェントはボールの動きに合わせてパドルを操作する必要がある。以上のことから、エージェントは行動によりパドルの移動先を正しく認識するほど、また、パドルの移動先とボールの位置を正確に認識するほど、適切な行動を選択できると考えられる。

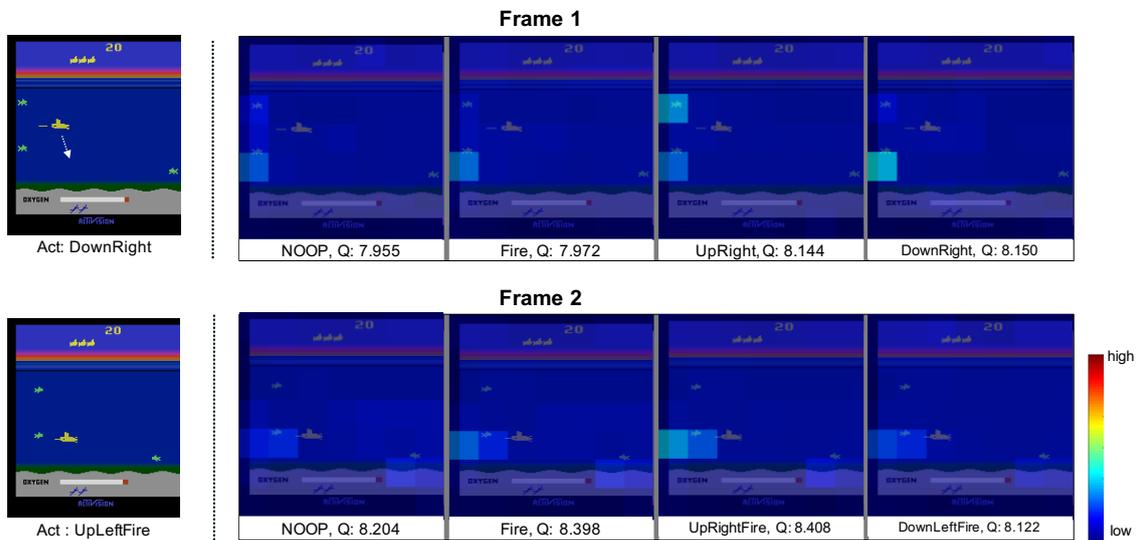
Seaquest. Seaquest は、ダイバーを救出しながら敵潜水艦や魚を撃破していくゲームである。エージェント (= 潜水艦) の行動は次の 18 つである：Fire, Up, Down, Left, Right の組み合わせパターンと, Noop.

Encoder-attention (状態価値). Seaquest における魚と酸素ゲージに着目した encoder-attention の可視化例を図 5.7a に示す。図 5.7a 上のシーンでは、エージェントは全てのフレームにおいて魚を注視し、一方で下のシーンでは酸素ゲージを注視していることが確認できる。Seaquest では、魚を撃破することでスコアを獲得でき、酸素残量がなくなるとゲームオーバーとなるため、エージェントは一定時間ごとに酸素を補給する必要がある。以上のことから、エージェントは魚を Seaquest の重要なオブジェクトとして認識し、同時に酸素補給の必要性和酸素補給のタイミングを正しく認識できていると考えられる。

Decoder-attention (advantage). Seaquest における decoder-attention の可視化例を図 5.7b に示す。Frame 1 から、エージェントは UpRight の行動を選択する場合は左上の魚、DownRight の行動を選択する場合は左下の魚を注視していることが確認できる。Seaquest では、エージェントの UpRight は潜水艦を右上に動かし、DownRight は右下に動かす。つまり、エージェントの行動選択による潜水艦の対する上下への移動方向は注視している魚の位置と一致する。したがって、エージェントは移動方向にいる魚を強く注視していると考えられる。Frame 1 で DownRight が選択された後、右下に移動した Frame 2 では攻撃を含む UpLeftFire が選択されている。つまり、Frame 1 でエージェントが注視していた魚を Frame 2 で攻撃していると言える。以上のことから、エージェントは魚を破壊すべき対象として認識し、それぞれの行動によって影響のある最も効率的な攻撃対象を正しく認識していると考えられる。



(a) Encoder-attention : 上は魚 (赤丸) に着目したシーンでの可視化例。下は酸素ゲージ (赤丸) に着目した可視化例で、酸素残量が枯渇し浮上するシーンである。これら可視化例は横方向に時系方向に並んでいる。



(b) Decoder-attention : attention 下のキャプションは行動名とその行動に対する Q 値を示し、入力画像下のキャプションは現状態でエージェントが選択した行動である。白矢印は潜水艦の進行方向を表している。また、Seaquest におけるエージェントが選択可能な行動の総数は 18 だが、4 つの行動をピックアップし示す。

図 5.7: Seaquest における attention weight の可視化例

まとめと考察. Encoder の出力を Value branch への入力として使用することで、encoder は状態価値に対する固有の attention weight を算出する。ここで、状態価値は現在の状態におけるリターンの期待値を表す。したがって、encoder-attention は全フレームにわたってスコアを得るために重要なオブジェクトを示していると考えられる。Decoder の出力を Advantage branch への入力として使用することで、decoder はエージェントの行動に対する attention weight を計算する。本手法では、decoder は action query を用いて、エージェントが選択した行動のみだけでなく、選択可能な行動それぞれに対して固有の attention weight を計算する。ここで、advantage は行動価値から状態価値を引いたものであり、行動そのものに対する価値である。したがって、decoder-attention は各 query (つまり各行動) に対して、その行動が選択された場合に影響を受けるオブジェクトや領域を示していると考えられる。

5.3.2 ロボット制御タスク

本節では、屋内環境におけるロボットの自律移動タスクを用いた実験について述べる。本実験では、Human Support Robot (HSR) [152] を用いた屋内環境における自律移動タスクに焦点を当てる。HSR は、トヨタ自動車が開発した介助・自立・生活面の幅広いサポートを目指している生活支援ロボットである。人間と同じ生活空間での動作を目的としているため、HSR の自律制御において信頼性は非常に重要な要素である。そこで、AQT によるロボットエージェントの自律移動性能に対する評価と、attention weight 可視化によるロボットエージェントの意思決定に対する分析について述べる。また、シミュレーション環境で学習したロボットエージェントを実環境へ適用し、AR インタフェースを用いた視覚的フィードバックの有効性についても評価する。

本実験における AQT のベースラインモデルとして、代表的な Value ベースな深層強化学習アルゴリズムである Deep Q-Network (DQN) [25] を採用する。比較した手法は、ベースラインモデルである DQN と AQT である。各モデルの学習終了は、学習ステップ数が 1.0×10^6 ステップとした。またエピソードの終了条件は 100 ステップ経過とした。

評価方法は以下の 3 つである。

- 屋内環境におけるロボットの自律移動性能に対する評価
- attention weight による視覚的説明
- AR を使ったユーザへの視覚的フィードバック

■ 学習環境とドメインギャップの解消

ロボット制御におけるオンライン強化学習では、実空間におけるロボットと環境間のインタラクションが困難であるため、シミュレーション環境を用いることが一般的である。本実験では、シミュレーション環境として Unity3D [153] を用いて歩行中の人間や家具などの複数の障害物を含む屋内

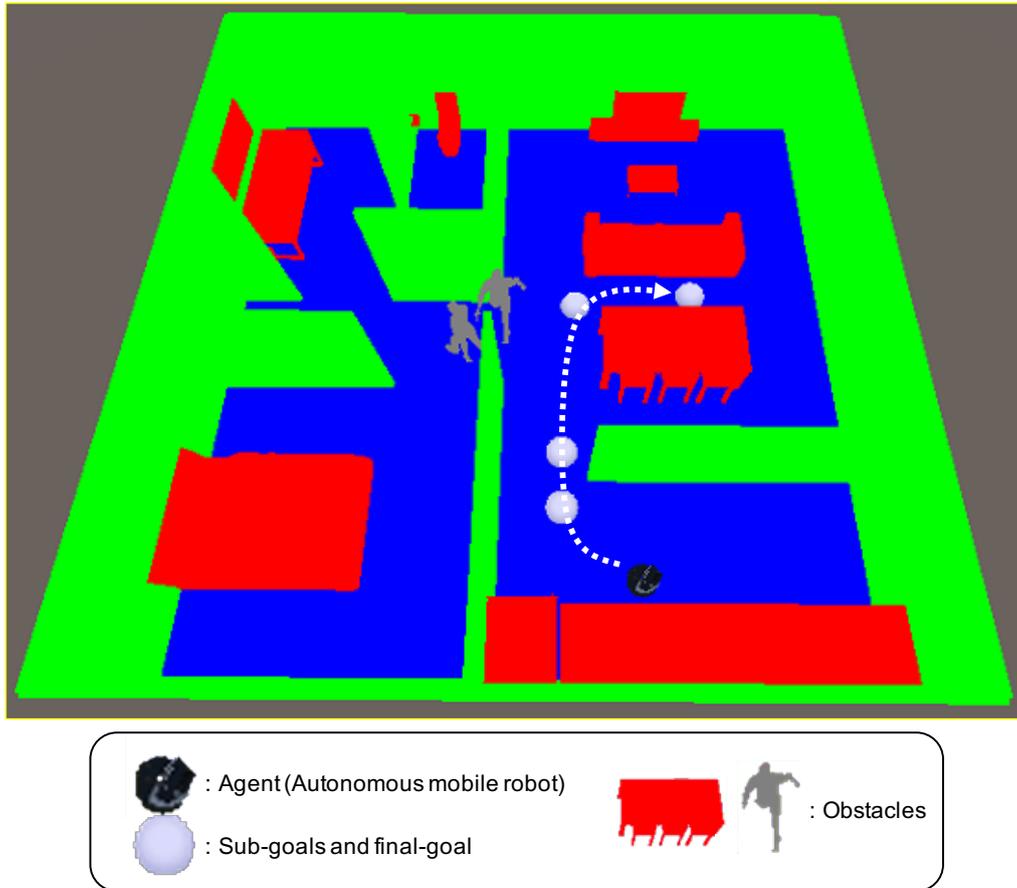


図 5.8: Unity で作成された屋内環境におけるロボット制御シミュレーション。

CG 環境を作成した (深層強化学習プロセスの実装は Unity ML-Agents Toolkit を用いる)。ここで、シミュレーション環境で学習したロボットエージェントを実世界に適用する際、入力情報のドメインギャップによる性能低下が課題として挙げられる。これは、実環境下で得られる画像などのセンサ情報と、CG 環境によりシミュレートした画像などのセンサ情報間の乖離に起因するものである。そこで、本実験ではこのドメインギャップを低減するために、エージェントモデルへの入力情報の中間表現としてセグメンテーション画像を採用する。

シミュレーション内のテクスチャは、図 5.8 のようにオブジェクトごとにテクスチャを光と影の影響を受けない単色のテクスチャに置き換え、セマンティックセグメンテーションの効果を再現する (ソファや机などのオブジェクトは赤色、床は青色、壁は緑色)。実世界で動作させる際は、ロボットのカメラから取得した RGB 画像を入力とするセマンティックセグメンテーションモデルにより、セグメンテーション画像を獲得する。このセグメンテーション画像をエージェントモデルへの入力とすることで、テクスチャの違いに起因するドメインギャップを低減する。本実験におけるセマンティックセグメンテーションモデルには、実環境データで十分に学習した SegNet [154] を採用する。

本実験における自律移動タスクの目的は、ロボットが障害物を避けながら最終ゴールに効率よく到達することである。このシミュレーション環境では、スタート位置と最終ゴール位置は各エピソード

Query branch in Robot control task

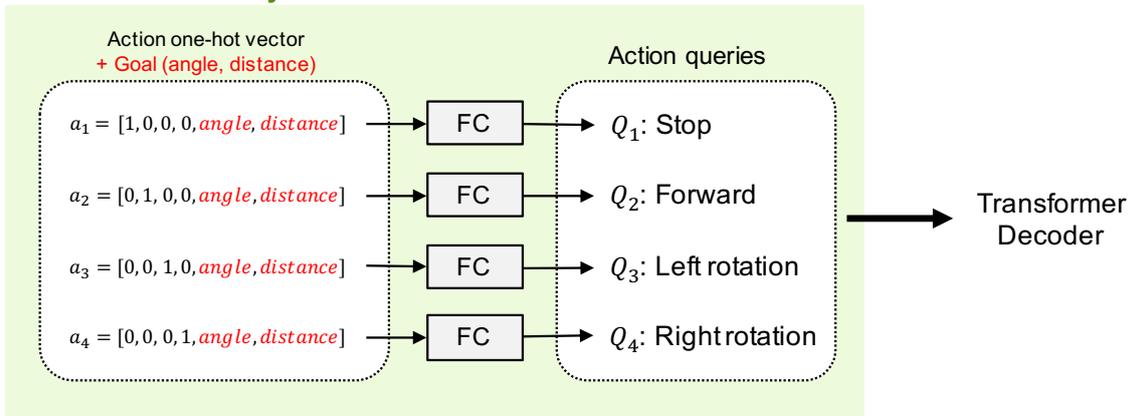


図 5.9: ロボット制御タスクにおけるゴール情報を考慮した Query branch.

ごとにランダムに設定され、スタートとゴール間にいくつかのサブゴールが自動的に生成した。ロボットエージェントの行動は停止、前進、左回転、右回転の4つの離散的な行動とした。このタスクを解決するために以下の報酬を設計した。各報酬 $reward$ を式 (5.4) に示す。

$$reward = \begin{cases} 10 & (\text{reach the goal}) \\ -0.01 & (\text{not getting closer to goal}) \\ -3 & (\text{collision with something}) \end{cases} \quad (5.4)$$

ロボットエージェントとゴール位置間の距離が 0.25 m 未満の場合、エージェントに正のゴール到達報酬 (+10) が与えられる。ロボットエージェントがゴールに近づいていない場合、ゴールに近づくように促すために微小な負の報酬 (-0.01) を与える。またロボットエージェントが何かしらのオブジェクトに衝突した場合は、障害物の回避行動を促すために負の衝突報酬 (-3) が与えられる。本実験におけるエージェントが受け取る報酬は、これら3種類の報酬の合計である。

■ ロボットエージェントモデルの詳細

DQN は行動価値を出力するモデル構造であるため、本実験における AQT の action branch は 5.1.3 節で述べた構造と同様のモデル構造とする。ここでロボットエージェントの行動は4つのため、action branch では4つの Q 値を出力する。

本実験では、設定したゴールへのロボットエージェントの自律移動を目的とする。そのため、エージェントモデルにはロボットのカメラ画像から取得するセグメンテーション画像と、追加情報としてゴール情報を与える必要がある。そこで、AQT モデルの action query にゴール情報を付加することで、ゴール情報を考慮した行動選択を促す。ここでゴール情報とは、エージェントからゴールまでの距離と角度情報である。本実験におけるゴール情報を付加した action query を図 5.9 に示す。ロボットエージェントの動作は停止/前進、左旋回/右旋回の4つであるため、各行動の One-hot vector

表 5.1: 100 エピソード間におけるタスク達成率

Agent model	Deep Q-Network	Action Q-Transformer (Ours)
Task accomplishment rate [%]	77	83

を定義する。これらの One-hot vector に、ロボットとゴールの相対的な角度と距離を表すゴール情報と結合する。角度情報は、ゴールがロボットに対する相対角度に比例する 0 から 1 の連続値で表される。ゴールがロボットの真後ろにある場合、角度値は 0 であり、ゴールが時計回りに移動すると角度値が増加する。距離情報は、他の情報とスケールを統一するため、ロボットとゴール間の実距離の 10 分の 1 に設定する。結合後の各ベクトルは、全結合層により特徴ベクトルに変換され、Decoder の Query (Action query) として使用する。

■ 屋内環境におけるロボットの自律移動性能に対する評価

本節では、ロボットの自律移動タスクにおける AQT モデルの制御性能について述べる。AQT の自律移動性能をシミュレーション環境で評価するため、100 エピソード間のタスク達成率により比較する。ここでのタスク達成条件は、ロボットが 100 ステップ以内に最終ゴールに到達し、衝突回数が 6 回以下となることである。ベースラインモデルである DQN のモデル構造は、3 つの畳み込み層と 3 つの全結合層から構成する。入力は AQT モデルと同様にロボットのカメラ画像から取得するセグメンテーション画像である。ゴール情報は、全結合層で特徴ベクトルに変換され、畳み込み層の出力と連結する。

表 5.1 に 100 エピソード間のタスク達成率を示す。表 5.1 から、AQT は DQN よりも 6% 高い、80% 以上の高いタスク達成率である。このことから、Transformer 構造を導入することで、AQT モデルはロボットの自律移動タスクにおいても高い制御性能を獲得していることが確認できた。

■ attention weight による視覚的説明

本節では、ロボットの自律移動タスクにおけるロボットエージェントの意思決定に対する AQT の有効性について述べる。シミュレーション環境におけるロボットエージェント (AQT モデル) の decoder-attention を可視化し、AQT の有効性を評価した。本実験における AQT の decoder には、行動の One-hot vector とゴールの位置情報を query として入力する。そこで、ゴールの位置を変化させ行動ごとの decoder-attention が適切に変化するか確認した。図 5.10 (a),(b) は、それぞれゴールが 2m 前方の左側 45 度と右側 45 度の位置であるときの attention weight を示している。ここで、(a) と (b) はどちらも同じ入力画像であり、ゴール位置のみ変化した可視化例である。以下では、これらの図 5.10 に示された attention weight を用いたロボットエージェントの意思決定について解析する。

図 5.10 (a) から、Left rotation を選択する場合は画像左側に位置する家具を注視し、Right rotation

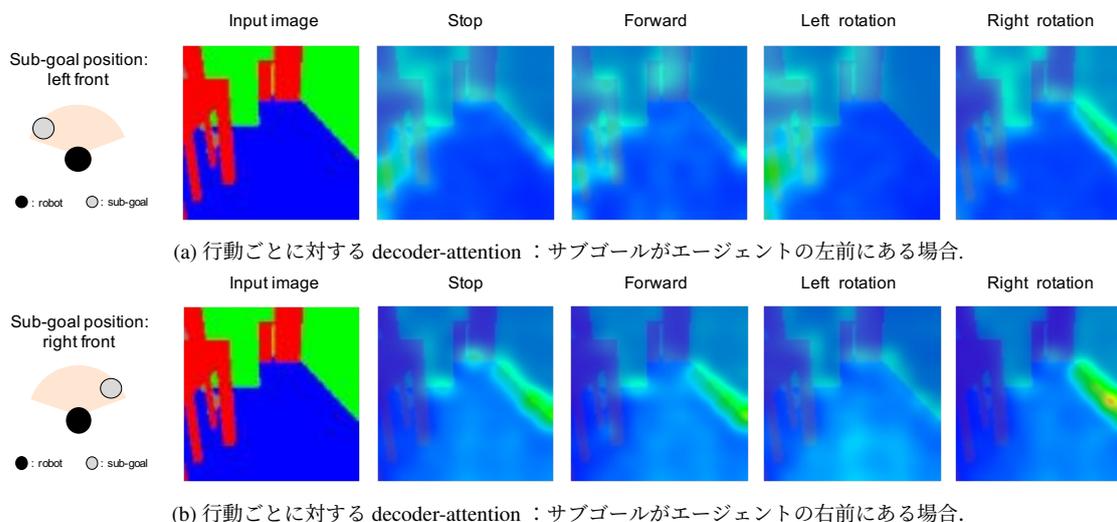


図 5.10: ロボットの自律制御タスクにおける decoder-attention の可視化例. 入力画像は (a),(b) とともに同じである.

を選択する場合は画像右側の壁を注視していることが確認できる. 一方で, Stop は現フレームにおいて周囲に影響を及ぼす行動ではないため, 画像上の広範囲に注意を向けている. したがって, ロボットエージェントは選択する行動ごとに影響のある領域やオブジェクトを注視していると考えられる. 図 5.10 (a), (b) から, ゴール位置が左前方から右前方に変化すると, ロボットエージェントの注視対象も変化していることが確認できる. Stop と Forward では, ロボットエージェントの注視対象は左側の家具から右側の壁へと移動していることが分かる. また Left rotation では, 左側の家具への注視が減少し, Right rotation では右側の壁への注視が大幅に増加する. したがって, ロボットエージェントはゴール位置に近いオブジェクトを注視し, 選択する行動が注視対象に近づく行動の場合, オブジェクトをより強く注視していることが分かる.

これらの結果から, decoder-attention は各行動とゴール位置に対するロボットエージェントの注視領域を正しく示しており, attention weight の可視化はユーザがロボットの判断を理解するのに役立つと考えられる.

■ AR を使ったユーザへの視覚的フィードバック

本節では, AR インターフェースを用いた視覚的フィードバック方法の有効性について述べる. 本実験では, AR インターフェースを用いた AQT モデルの decoder-attention 可視化による視覚的フィードバックの有効性を, ロボットエージェントの動作に関するユーザの理解度にもとづいたアンケート調査を実施する. 調査方法は以下の 2 つである.

- AR を用いた視覚的フィードバックの有無によるロボット動作の予測設問に対する正答率.
- ロボット動作の理解に関する主観的なアンケート調査.

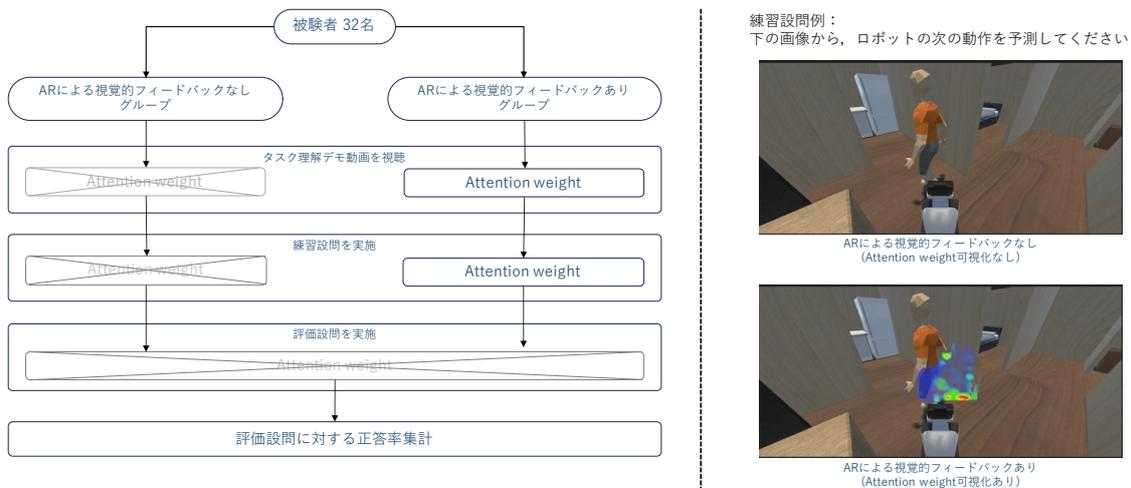


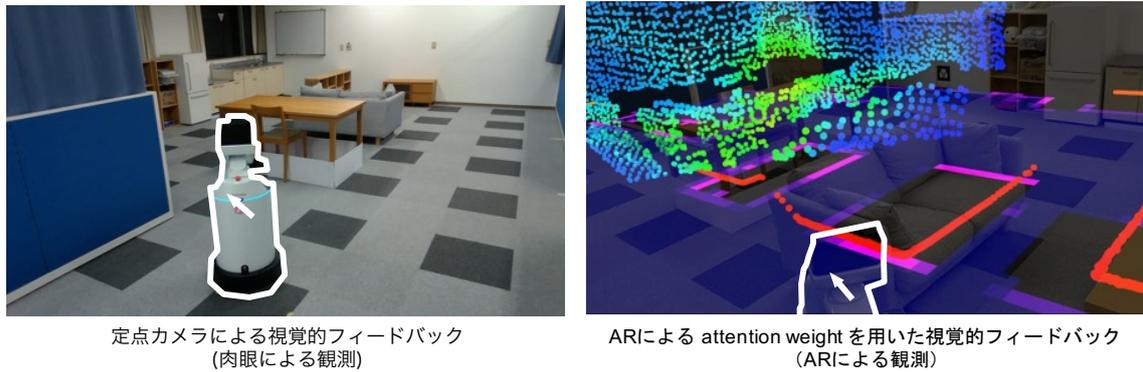
図 5.11: ロボット動作に対する予測設問に関する調査のフローチャート。右に練習設問例を示す。

表 5.2: ロボットの振る舞いに対する評価設問の平均正答率 [%]

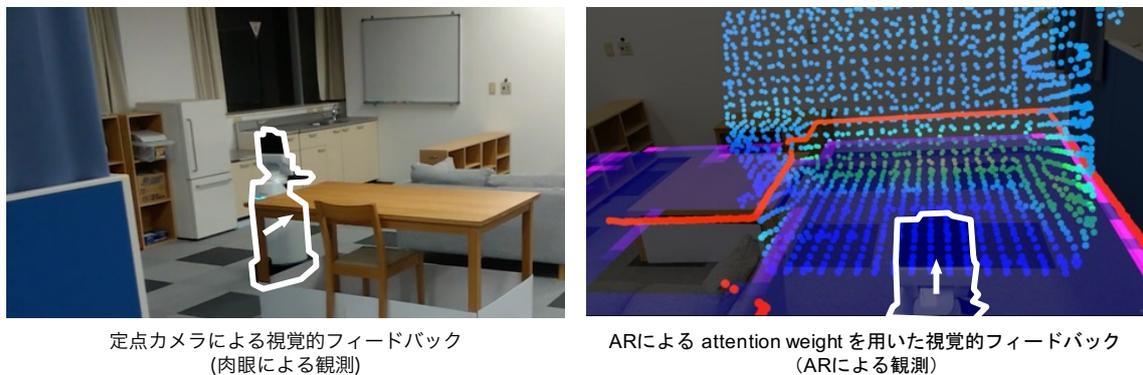
被験者グループ	AR による視覚的フィードバックなし	AR による視覚的フィードバックあり
平均正答率	31.8	38.8

AR を用いた視覚的フィードバックの有無によるロボット動作の予測設問に対する正答率。 AR による attention weight の視覚的フィードバックの有無によって、被験者 32 名を 2 グループに分け、ロボットの振る舞いを予測できるかを調査した。本調査では、ロボットの振る舞いに対する網羅的な評価を行うため、図 5.4 (a) に示すシミュレーション環境を用いた。図 5.11 は本調査のフローチャートを示しており、以下に調査手順の詳細を述べる。

1. 被験者は、ロボットが自律移動タスクを実行しているデモ映像を視聴する。ここでデモ映像は、被験者のグループによって異なる映像を提供する。AR による視覚的フィードバックありのグループには、AR によって attention weight を可視化したデモ映像を提供し、もう一方のグループには attention weight なしの通常のデモ映像を提供する。
2. 被験者は、提供した静止画像からロボットの振る舞いを予測する 10 問の練習設問に回答し、ロボットの振る舞いに対する理解を促進する。ここで静止画像は、1 で提供したデモ映像と同様にグループごとの方法で可視化された画像である。つまり、AR による視覚的フィードバックありのグループは、実空間上におけるロボットの注視領域も含めてロボットの振る舞いを理解できる。
3. 被験者は、通常の静止画像（attention weight 可視化によるフィードバックなし）にもとづいて、ロボットの振る舞いを予測する 10 個の評価設問に答える。これらの評価設問に対する被験者の正答率を集計し、AR による視覚的フィードバックの有効性を調査した。



(a) HSR が左回転するシーン.



(b) HSR が前進するシーン.

図 5.12: 実世界にロボットエージェントの振る舞いに対する視覚的フィードバック例. 左に定点カメラによる動画例, 右に AR デバイスを用いた attention weight 可視化動画例を示す. また, 白枠で囲われた領域には HSR が存在し, 白矢印は HSR の進行方向を示している.

全被験者の回答に対する平均正答率を表 5.2 に示す. AR による視覚的フィードバックありグループは, なしグループと比較し, 全参加者の平均正答率が 7% 向上していることが確認できる. この結果は, AR による視覚的フィードバックがロボットの振る舞いを予測する上でユーザにとって効果的に役立つことを示唆している. 言い換えれば, AR による視覚的フィードバックはロボットの振る舞いに対するユーザの理解を深める効果があることを示している.

ロボットの振る舞いに対するユーザの主観的理解. AR による attention weight を用いた視覚的フィードバックが, ロボット動作に対する被験者の主観的理解度を向上させるか評価するため, ユーザに対してアンケート調査を実施した. 本調査は, 実世界での AR を用いた attention weight 可視化による視覚的フィードバックの有効性を調査するため, 図 5.4 (b) に示すような実世界における実験に取り組んだ.

調査方法を以下で説明する. 23 人の被験者に, 実世界上でロボットが自律的に動作する 2 種類の動画を視聴してもらった. 1 つは定点カメラに用いた動画 (肉眼による観察), もう 1 つは AR デバイスを用いて attention weight を可視化した動画 (AR による観察) である. それぞれの動画例を図 5.12

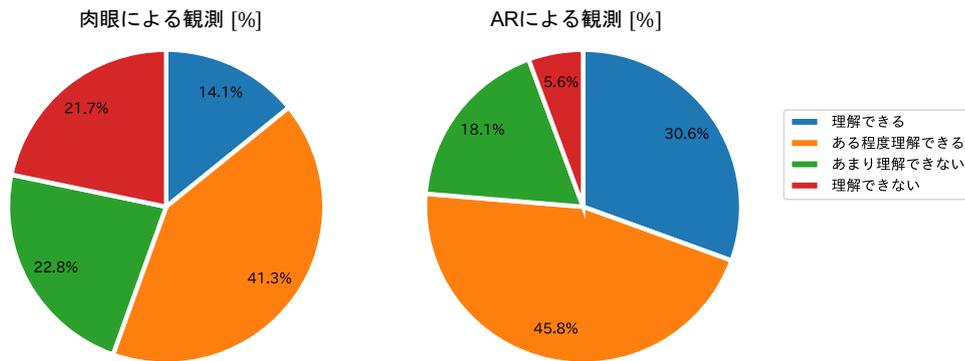


図 5.13: ロボットの振る舞いに対するユーザの主観的理解度アンケートの回答結果. 各数値は回答者の割合を表す.

に示す. 動画視聴後に, ロボットの振る舞いに対する理解度を問うアンケートを実施した. このアンケートでは, 理解度を「理解できる」, 「ある程度理解できる」, 「あまり理解できない」, 「理解できない」の4段階で被験者に回答してもらう. 動画は, 実世界の異なる4シーンで, Microsoft社が開発したARデバイス HoloLens2 [155] を用いて撮影した. 各動画に対する理解度アンケートの調査結果を図 5.13 に示す. 肉眼による観察では, 「理解できる」もしくは「ある程度理解できる」と回答した被験者の割合が56%であることが確認できる. これに対し, ARによる観察は76.4%と20.4%増加していることが分かる. また, 「理解できない」と回答した被験者の割合も, 肉眼による観測では21.7%であったのに対し, ARによる観測では5.6%と15.4%減である. これらの結果から, ARを用いた attention weight による視覚的フィードバックは, ロボットの振る舞いに対するユーザの主観的な観点においても理解を深めるのに有効だと考えられる.

5.4 まとめ

本章では, 深層強化学習エージェントの意思決定プロセスに対する解釈性向上を目的とし, 行動情報を query として使用する Transformer encoder-decoder 構造にもとづく XRL 手法 Action Q-Transformer (AQT) を提案した. AQT の decoder は, 行動情報を query として表現した action query を用いて, 行動ごとに固有の attention weight を獲得する. これにより, エージェントが選択可能な全ての行動に対する attention weight を獲得することができ, エージェントモデルの意思決定に対する詳細な分析を可能とした. さらに, 実空間上で動作するロボットエージェントに対するユーザの理解促進を目的とし, ユーザがエージェントを直感的かつ即時的に理解できる枠組みである Augmented Reality (AR) を用いたユーザに対する視覚的フィードバック手法を提案した. 本手法は, 実空間上におけるロボットエージェントモデルの意思決定に対する attention weight を AR を用いて可視化することで, ユーザが AR デバイスを介してエージェントの行動選択に対する判断根拠を容易に解釈可能とした.

Atari2600 のビデオゲームを用いた実験では, Rainbow をベースとした AQT を用いることで, encoder

では状態価値に関する attention weight, decoder では行動ごとに対する固有の attention weight を獲得できることを確認した. この実験から, decoder の attention weight は行動ごとに異なる注視領域を持つことを確認した. エージェントの意思決定を表すこれらの異なる attention weight を可視化することで, ビデオゲームタスクにおいてエージェントの意思決定に対する判断根拠を網羅的に分析できることを示した. さらに, transformer 構造に対するスケール則に対する対処方法として, Target Trained Q-network (TTQ) を提案し, 一部ビデオゲームタスクにおいて, ベースラインの rainbow よりも高いスコアを達成することを確認した. また, これらの実験は屋内環境においてロボットの自律移動タスクにおいても同様に確認した. ロボットの自律移動タスクでは, AR を用いた視覚的フィードバック手法に対する評価として, ユーザによるロボットエージェントの振る舞い予測とユーザの主観的理解に関するアンケート調査を実施した. 調査結果より, AR を用いた視覚的フィードバックにより, ロボットエージェントの振る舞い予測設問に対するユーザの正答率が向上することを確認した. さらに, 主観的理解に対するアンケート調査では, 定点カメラによる教示より, 「理解できる」と回答したユーザが大幅に向上したことを確認した. 本研究は, 深層強化学習エージェントの意思決定プロセスに対する視覚的説明の一種である. これらの視覚的説明に関する研究では, 可視化した視覚的説明情報から深層強化学習エージェントモデルの行動選択に対する洞察を得る. そのため, この最終的な洞察はユーザに主観に依存しており, ユーザごとの解釈が異なってしまうケースが発生してしまう危険性がある. そこで, エージェントモデルの意思決定プロセスを自然言語にもとづく文章として説明する手法を開発が課題として挙げられる.

第6章

視覚的説明から言語的説明への拡張

本章では、深層強化学習エージェントの意思決定プロセスに対する視覚的説明から言語的説明への拡張について議論する。4, 5章では、深層強化学習エージェントの意思決定プロセスに対する視覚的説明手法について述べ、ビデオゲームやロボット制御での評価実験から、エージェントモデルの行動選択に対する判断根拠を視覚的に解析できることを示した。これらの視覚的説明手法は、ユーザに対してエージェントモデルの注視領域を画像などの視覚情報として提供できる。しかし、最終的なエージェントモデルの行動選択に対する洞察は、ユーザがその視覚情報から読み取る必要があるため、ユーザの主観に依存してしまう。そこで、これら視覚的説明手法とは異なり、エージェントモデルの意思決定を自然言語による説明/解釈を提供する言語的説明が注目を集めている。エージェントモデルに対する言語的説明手法は、自然言語による文章形式でユーザに対して説明を提供するため、エージェントモデルの意思決定プロセスに対する洞察がユーザの主観に依存しにくい。また自然言語処理分野における大規模言語モデル (Large Language Model; LLM) [16, 17, 18] の登場により、言語モデルに大幅な技術進歩が起きていることから、エージェントモデルの判断根拠に対する言語的説明の期待が高まっている。

説明可能な強化学習 (eXplainable Reinforcement Learning; XRL) の文脈における言語的説明として、テンプレート文を用いた説明文生成アプローチ、自由形式での説明文生成アプローチがある。テンプレート文を用いた説明文生成アプローチでは、あらかじめ質問と回答のテンプレート文を人手によって複数用意し、ユーザからの質問とエージェントの状態や行動などの要素から、テンプレート文にもとづき説明文を生成する [14]。このアプローチは、テンプレートしたがって説明文を生成するため、ユーザにとって理解しやすい文章を提供することができる。しかし、説明できるシーンが非常に限定的であり、多様な意思決定を行う深層強化学習エージェントモデルに対する網羅的に洞察には限界がある。一方で、自由形式での説明文生成アプローチでは、言語モデルなどを用いてエージェントモデルの意思決定に対する説明文を生成する [15, 123]。そのため、エージェントモデルの意思決定に対し、明瞭さを犠牲にすることなく、より表現力豊かな洞察を生成できる可能性がある。

本研究の目的は、深層強化学習エージェントモデルの意思決定に対する判断根拠をユーザが理解可能な自然言語の文章形式として提供することである。そこで、エージェントモデルに対する視覚的説明手法にもとづき、視覚的説明情報に利用した LLM による言語的説明手法への拡張を提案する。本研究におけるエージェントモデルは、4, 5章で述べたような視覚的説明手法を用いて構築する。これらの視覚的説明情報やエージェントモデルが選択した行動情報を LLM のプロンプトとすることで、エージェントモデルの意思決定に対する判断根拠を LLM によって言語化する。

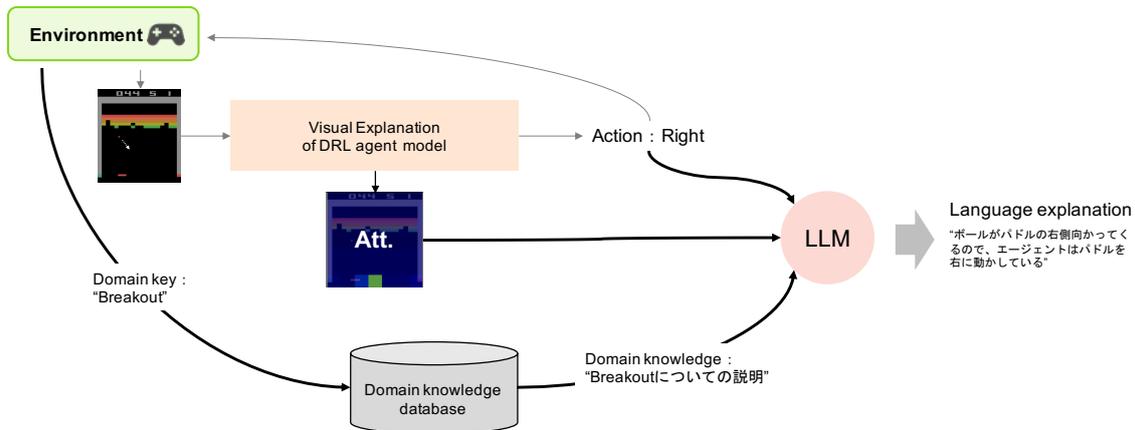


図 6.1: 深層強化学習エージェントモデルの意思決定に対する言語的説明手法の概略

本章の構成は次の通りである。6.1 節では、視覚的説明情報を活用した LLM による言語的説明手法について述べる。6.2 節では、6.1 節で述べた言語的説明手法の有用性を確認するため、ビデオゲームタスクである Atari2600 を用いた評価実験について述べ、言語的説明手法の活用についても議論する。最後に、6.3 節で本章をまとめる。

6.1 視覚的説明情報を活用した LLM による言語的説明

本節では、4,5 章で述べた深層強化学習エージェントに対する視覚的説明情報を活用した大規模言語モデル (LLM) による言語的説明手法を提案する。深層強化学習エージェントモデルの意思決定に対する言語的説明手法の概略を図 6.1 に示す。図 6.1 に示す通り、本手法は視覚的説明手法を導入したエージェントモデル、説明文生成 LLM、Domain knowledge database で構成する。

本手法では、エージェントモデルに対する視覚的説明情報と、エージェントが選択した行動情報、ドメイン知識情報を LLM への入力プロンプトとし、LLM がこれら情報にもとづき、エージェントモデルの振る舞いに対する説明文を生成する。ここで視覚的説明情報は、Mask-attention A3C の場合は mask-attention (4 章参照)、Action Q-Transformer の場合は decoder の attention weight (5 章参照) など、エージェントモデルの方策に直結する注視領域を指す。本研究では、説明対象ドメインを Atari2600 のビデオゲームタスクとし、言語的説明に用いる LLM として ChatGPT4o を採用する。ChatGPT4o は、OpenAI によって開発された多言語対応かつマルチモーダルな Generative Pre-trained Transformer (GPT) [17] である。前身である GPT と比較し、モデルサイズと学習データの大規模化、入力可能なトークン数の大幅な増加、テキストだけでなく画像も入力可能、Reinforcement Learning from Human Feedback (RLHF) [10] を用いたアライメントによる柔軟な対話性能の獲得などの特徴がある。この ChatGPT4o は、supervised fine-tuning 等によって説明対象ドメインへ特化させず、公開されたモデルを利用し、説明対象ドメインの知識は外部データベースを活用することで補完する。

6.1.1 Domain knowledge database

言語的説明に用いる LLM は、説明対象ドメインに対する supervised fine-tuning 等を行わないため、説明対象ドメインに関する知識を有していない。しかし、エージェントモデルの意思決定プロセスに対する説明文を生成するには、説明対象のドメイン知識が必要不可欠である。そこで、エージェントモデルの制御目的であるドメイン由来の知識を外部データベースにより補完する。本研究では、Atari2600 のビデオゲームタスクを目的とするため、この外部データベースには Atari2600 のビデオゲームに対するドメイン知識を格納する。ここでのドメイン知識とは、それぞれのビデオゲームに関する概要などが記述されたテキスト情報とする。格納するドメイン知識は、次の web サイトからゲームごとにスクレイピングすることで収集する：OpenAI gym のドキュメントサイト¹、AtariAge²。OpenAI gym のドキュメントサイトには、ゲームの簡単な説明、およびエージェントが選択可能な行動や報酬設計について記載されている。AtariAge には、ゲームのルールなどを始めとしたゲームのプレイ方法などについて記載されている。これら 2 つの web サイトは、ノイズや虚偽の情報がない信頼できる情報源である。

6.1.2 LLM による説明文生成のためのプロンプトエンジニアリング

エージェントモデルの言語的説明に向けた LLM へのプロンプトとして、LLM への指示文に加え、視覚的説明情報とエージェントモデルが選択した行動情報、Domain knowledge database から参照したドメイン知識が記述されたテキスト情報を用いることで、LLM からエージェントモデルの意思決定に対する説明文を得る。以下で、LLM へ提供する情報について述べる。

視覚的説明情報。 4, 5 章で述べた Mask-attention A3C や Action Q-Transformer などの視覚的説明手法を導入したエージェントモデルから取得できる視覚的説明情報を LLM に提供する。ここで提供する視覚的説明情報は、エージェントモデルへの入力であるゲーム画面 (RGB 画像) と、その入力画像に対して attention weight 等を重畳させたエージェントモデルの注視領域を示すマップとする。

エージェントモデルの行動情報。 本研究では、Atari2600 のビデオゲームタスクを対象とする。そのため、エージェントモデルの行動情報として、Atari2600 における離散的な行動空間のサブセット名 (テキスト情報) を行動情報とする。ここで、サブセット名は NOOP や FIRE, UP, LEFT など Atari2600 のゲームをプレイする際のコントローラの制御名である。

ドメイン知識情報。 本手法では、ドメイン知識をテキスト情報としてプロンプトに記述する形で、LLM が保有していないドメイン固有の知識を補完する。プロンプトに記述するドメイン知識情報は、6.1.1 節で述べた Domain knowledge database から参照する。

¹<https://ale.farama.org/environments/>

²<https://atariage.com/index.php>



図 6.2: ステップ・バイ・ステッププロンプトの概要。

■ ステップ・バイ・ステッププロンプト

複雑なタスクを一度のプロンプトで指示するのではなく、段階的にステップ・バイ・ステップで指示することで、LLM の回答精度が向上することが報告されている [156]. そこで、上述の視覚的説明情報等から言語的説明に繋げるために、与えられた情報から直接的に説明文を生成するのではなく、逐次的に指示を行う。ステップ・バイ・ステッププロンプトの概要を図 6.2 に示す。エージェントモデルの判断根拠に対する言語的説明において、視覚的説明情報からエージェントモデルがどこを注視しているか、また説明対象ドメインにおいてどのフレームがキーとなるかを抽出することが重要である。本プロンプトでは、LLM とユーザの対話形式にて、重要イベントの抽出と、言語的説明の 2 段階に分け、エージェントモデルの意思決定に対する説明文を得る。重要イベント抽出では、ユーザから与えられた情報（視覚的説明情報、エージェントモデルが選択した行動情報、ドメイン知識など）から、説明対象ドメインにおける重要なイベントと、そのイベントにおいてエージェントが何に注視しているかを抽出する。そして、言語的説明では、重要イベント抽出時に抽出した説明対象に対し、エージェントの意思決定に対する判断根拠の説明文を生成する。

■ 短期説明と長期説明

エージェントモデルの判断根拠に対する網羅的な説明文生成として、短期説明と長期説明の 2 種類のプロンプトを提案する。各プロンプトの例を図 6.3 に示す。短期説明プロンプトは、ユーザが指定した数ステップ分の短期的な情報から、エージェントモデルの判断根拠に対する説明文を生成する。ここで、ユーザにより提供される視覚的説明情報は、2 つの画像（エージェントモデルへの入力である RGB 画像、注視領域を示したマップ）を並列に連結し、1 枚の画像データとして LLM に入力する。短期的説明では、この画像データを複数ステップ分作成し、この複数枚の画像データをプロンプトとして LLM に提供する。一方で、長期説明プロンプトは、ユーザによって 1 エピソードや特定範囲の長期的な情報を提供し、エージェントモデルの判断根拠に対する説明文を生成する。ここで、ユーザにより提供される視覚的説明情報は、短期的説明と同様に、2 つの画像（エージェントモデルへの入力である RGB 画像、注視領域を示したマップ）を並列に連結し画像データとするが、これらの画像データを単一の動画データとして LLM に入力する。

本プロンプトは、システムプロンプトと、上述の重要イベント抽出プロンプトおよび、言語的説明プロンプトで構成されている。以下で、それぞれのプロンプト詳細について述べる。

システムプロンプト. プロンプトの始めには、LLM に取り組むタスクについての指示文を記述

短期説明：説明対象の数 RLstep 分を人手により指定

システムプロンプト

特定のゲームをプレイするエージェントがいます。あなたは、提供された動画からエージェントによる判断根拠を言語化するシステムです。
提供される情報は以下の通り。
【動画】 (video) : 説明のベースとなる情報です。画面左はゲーム画面であり、画面右はエージェントがゲーム画面のどこを注視していたかを示しています。ここで、エージェントの注目領域がplayer mapが示されています。アタリは赤い点を示しています。
【行動リスト】 (actions.txt) : フレームごとに選択したエージェントの行動が、テキスト形式で記載されています。行動はカンマ区切りで、時系列順に記載されています。
【ゲームの説明】 (game.txt) : エージェントがプレイしているゲームに関する情報が提供されます。

重要イベント抽出

ユーザー: 提供された動画を解析し、重要なイベントを抽出してください。そして、抽出したイベントにおいて、エージェントが何を注視しているか、もしくは何を注視していないかを明らかにしてください。

LLM: 重要なイベントの抽出は〜
エージェントの注視は〜

言語的説明

ユーザー: 抽出した注視対象にもとづき、ゲームをプレイしているエージェントの意思決定に対する判断根拠を説明してください。説明する際は、注意事項を厳守してください。

LLM: エージェントの意思決定における判断根拠

長期説明：エピソードもしくは一定範囲を指定

システムプロンプト

特定のゲームをプレイするエージェントがいます。あなたは、提供された動画からエージェントによる判断根拠を言語化するシステムです。
提供される情報は以下の通りです。
【動画】 (video) : 説明のベースとなる10秒の動画情報です。画面左はゲーム画面であり、画面右はエージェントがゲーム画面のどこを注視していたかを示しています。ここで、エージェントの注目領域がplayer mapが示されています。
【行動リスト】 (actions.csv) : エージェントがフレームごとに選択した行動がcsv形式で記載されています。最初の行はヘッダーであり、列名はFrame (動画のフレーム番号) とActions (エージェントが選んだ行動) でカンマ区切りにより記載されています。
【ゲームの説明】 (game.txt) : エージェントがプレイしているゲームに関する情報が提供されます。

重要イベント抽出

ユーザー: 提供された動画を解析し、重要なイベントを抽出してください。そして、抽出したイベントにおいて、エージェントが何を注視しているか、もしくは何を注視していないかを明らかにしてください。

LLM: 重要なイベントの抽出は〜
エージェントの注視は〜

言語的説明

ユーザー: 抽出したイベントにもとづき、ゲームをプレイしているエージェントの意思決定に対する判断根拠を説明してください。説明する際は、注意事項を厳守してください。

LLM: エージェントの意思決定における判断根拠

図 6.3: 短期説明と長期説明プロンプトの例。

する。記述内容としては、提供した情報をもとづきエージェントモデルの意思決定に対する判断根拠を言語化するタスクであること、ユーザから提供する各情報に関してである。ここで、提供する情報は上述の「視覚的説明情報」、「エージェントモデルの行動情報」、「ドメイン知識情報」である。視覚的説明情報に関する記載は、短期説明プロンプトと長期説明プロンプトで異なるため、それぞれに合わせた形式を記載する。

重要イベント抽出。 ユーザから提供された情報（視覚的説明情報や、エージェントモデルが選択した行動情報、ドメイン知識など）から、説明対象ドメインにおける重要なイベントと、そのイベントにおいてエージェントが何に注視しているかを抽出する。

言語的説明。 重要イベント抽出時に抽出したイベントに対し、エージェントの意思決定に対する判断根拠の説明を指示する。また、説明する際の注意事項も同時に記述する。記述した注意事項は次の通りである。

- エージェントモデルの注視にもとづく説明文生成を促すための指示と、生成文のフォーマット指定。
例：「xxx フレームで xxx を注視しているから、エージェントは xxx の行動を選択した」というように、提供された動画からエージェントが何を注視しているかにもとづいて、エージェントの判断根拠を説明してください。
- エージェントモデルが必ずしもゲームタスクを攻略できているとは限らないことを考慮するための指示。制御失敗時に関するシーンも説明できるように促す。
例：エージェントは常にプレイに成功するとは限りません。エージェントがゲーム攻略に失敗した場合は、なぜ失敗したのかをエージェントの判断根拠をもとに説明してください。
- ユーザが容易に理解できるようにするための指示。
例：誰にでもわかりやすい言葉で説明してください。

- 視覚的説明情報以外は、説明文生成の補足として用いるための指示。
例：動画の他に、エージェントの行動リストやゲームの説明という形で補足情報が与えられ
ます。動画以外の情報は補足であることに注意してください。

6.2 評価実験

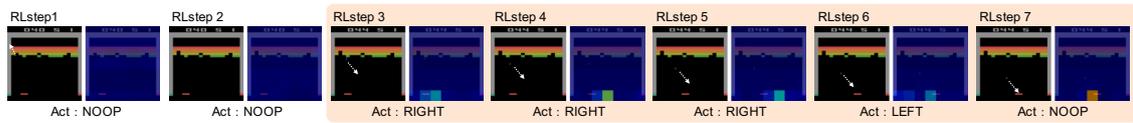
本節では、視覚的説明情報を活用した LLM による言語的説明手法の生成文が、ユーザの理解に有効であるか評価する。本実験では、Atari2600 の Breakout, Seaquest, Boxing, Space Invaders に焦点を当てる。エージェントモデルには、5 章で提案した Action Q-Transformer (AQT) モデルを用いる。AQT モデルによって取得する視覚的説明情報は、AQT モデルが選択した行動に対する decoder 部の attention weight とする。つまり、エージェントモデルの選択した行動と、その行動に対するエージェントモデルの注視領域を用いる。また、説明文生成に用いる LLM は ChatGPT4o とし、短期説明における視覚的説明情報は 7 ステップ分とした。

6.2.1, 6.2.2, 6.2.3 節では、Breakout, Seaquest, Boxing における言語的説明例を示し、エージェントモデルの意思決定プロセスに対して、妥当な説明文が生成できているか議論する。また、6.2.4 節では、前節で述べた深層強化学習エージェントの判断根拠に対する言語的説明の応用として、高性能な深層強化学習エージェントからユーザがノウハウを学ぶためのマニュアル作成を挙げ議論する。

6.2.1 Breakout における言語的説明

Breakout は、プレイヤーがパドルを操作してボールを打ち返し、画面上部のブロックを破壊するゲームである。エージェント (パドル) には 3 つの行動がある: NOOP, LEFT, RIGHT。ここで、NOOP は動作なしを意味する。Breakout における異なる 2 シーンの言語的説明例を図 6.4, 6.5 に示す。

図 6.4 は、パドルにボールが向かってくるシーンである。図 6.4 (a) から短期説明プロンプトでは、パドルの右側にボールが向かう RLstep 3-5 において、「エージェントはボールが右側へ落ちてくると予測し、パドルを右方向へ移動しました」とし、その根拠として「ボールの動きに対応したパドル右側のスペースに注視していることから明らかです」と説明している。RLstep 3-5 の attention weight では、パドルの右側を注視していることから、エージェントの注視対象を正しく説明している。したがって、パドル右側にボールが進行し、ボールを打ち返すという事実にも則しており、その根拠としてエージェントモデルの注視領域に着目した説明文が生成できている。また、パドルの位置を調整している RLstep 6,7 においても、LEFT を選択した根拠を「注視対象が左側の空間にシフトしていることから説明できます」とし、NOOP を選択した根拠を「パドルとボールの接触地点に注視しています」と説明している。この説明も attention weight から、RLstep 3-5 と同様に正しく注視対象を捉えていることが分かる。このようにエージェントモデルの注視領域を正確に述べた上で、エージェントモデルの振る舞いに対して説明できていることを確認した。図 6.4 (b) から長期説明プロンプトでは、視覚的説明情報として入力した動画データ等にもとづき、重要イベントとしてボールを打ち返



ChatGPT4oによる生成文（一部抜粋）

RLstep 3-5 (RIGHT)
 エージェントはボールが右側へ落ちてくると予測し、パドルを右方向へ移動しました。この判断は、ボールの動きに対応したパドル右側のスペースに注視していることから明らかです。
RLstep 6 (LEFT)
 エージェントは、パドルに対しボールの軌道が途中で左方向に変化したことを認識し、急いでパドルを左へ移動しました。この判断は、注視対象が左側の空間にシフトしていることから説明できます。
RLstep 7 (NOOP)
 ボールがパドルに非常に近づいているため、エージェントはパドルの位置が適切であると判断し、移動を止めました。この時点での注視対象は、パドルとボールの接触地点に注視しています。

(a) 短期説明プロンプト。上部は視覚的の説明情報とエージェントモデルの行動情報、下部は ChatGPT4o による生成文 (一部抜粋) を示し、オレンジ色でハイライトされたステップが該当する。

重要イベント抽出



ChatGPT4oによる生成文（一部抜粋）

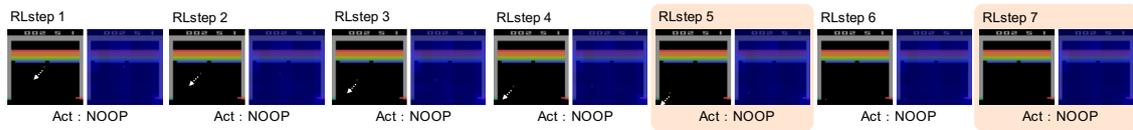
RLstep 13-15 : 行動「RIGHT」
 エージェントはパドルを右方向に移動させ、画面右下付近、つまりボールの移動先に注視していることが確認されました。エージェントはボールが右方向に動いているのを認識し、パドルをその位置に移動させることでボールを返すことを目的としているようです。ボールの移動先に注視が集中しているため、ボールの現在位置と予測位置をもとに行動を選択したと推測されます。この行動は適切で、エージェントはボールを正確に追跡していると判断されます。

(b) 長期説明プロンプト。上部は、重要イベント抽出にて抽出されたステップにおける視覚的の説明情報とエージェントモデルの行動情報、下部は ChatGPT4o による生成文 (一部抜粋) を示し、オレンジ色でハイライトされたステップが該当する。

図 6.4: Breakout におけるパドルがボールに向かうシーンに対する言語的説明例。ゲーム画面上の白矢印は、ボールの進行方向を示している。

すシーン (RLstep 13 - 17) を抽出している。ボールの打ち返しは、Breakout においてスコアに直結する行動であることから、重要なイベントを正しく抽出できていると考えられる。また、ChatGPT4o による生成文として「画面右下付近、つまりボールの移動先に注視していることが確認されました」と説明している。これは、短期説明 (図 6.4 (a) 参照) での生成文と同様であることから、エージェントモデルの注視領域を正確に述べた上で、エージェントモデルの振る舞いに対する説明できていると言える。

図 6.5 は、ボールの打ち返しに失敗したシーンである。このシーンは、学習途中の AQT モデルによって取得した視覚的の説明情報を用いた。図 6.5 (a) から短期説明プロンプトでは、ボールがパドル下に落ちる RLstep 5 において、「エージェントはボールとパドル周辺を注視しています」と注視対象がボールとパドルであるとし、「ボールを打ち損ねる可能性が高い」や「ボールの正確な軌道予測に失敗」など制御に失敗したと述べている。また、ボールが落下後の RLstep 7 においては、「注視が弱まり、画面全体に広がっています」と述べ、「ボールを打ち損ねた後の状態である可能性が高い」や「ミスを認識している状態だと考えられます。」と説明している。このシーンは、ボールの打ち返しに失敗したシーンであることから、制御に失敗している説明は適切である。一方で、attention

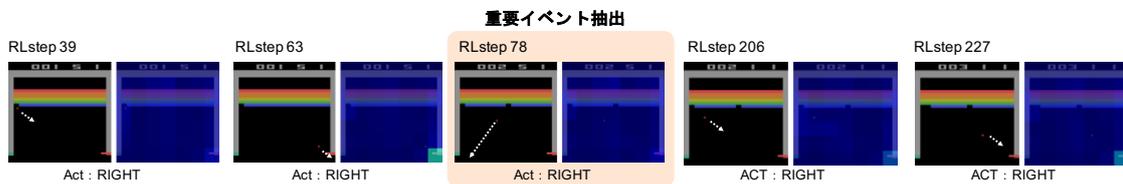


ChatGPT4oによる生成文（一部抜粋）

RLstep 5
 ボールがパドルの直前に到達。エージェントはボールとパドル周辺を注視しています。ここでは、エージェントがボールを適切に跳ね返すタイミングを逃してしまった可能性があります。「NOOP」を選択した結果、パドルの位置を調整できず、ボールを打ち損ねる可能性が高いです。この段階での判断ミスは、ボールの正確な軌道予測に失敗したことが原因と考えられます。

RLstep 7
 エージェントの注視が弱まり、画面全体に広がっています。ボールを打ち損ねた後の状態である可能性が高いです。エージェントの注視が分散しており、次の行動を考えているか、ミスを認識している状態だと考えられます。この結果、ゲーム攻略に失敗した理由としては、パドルの調整タイミングを逃したことが挙げられます。

(a) 短期説明プロンプト。上部は視覚的説明情報とエージェントモデルの行動情報，下部は ChatGPT4o による生成文 (一部抜粋) を示し，オレンジ色でハイライトされたステップが該当する。



重要イベント抽出

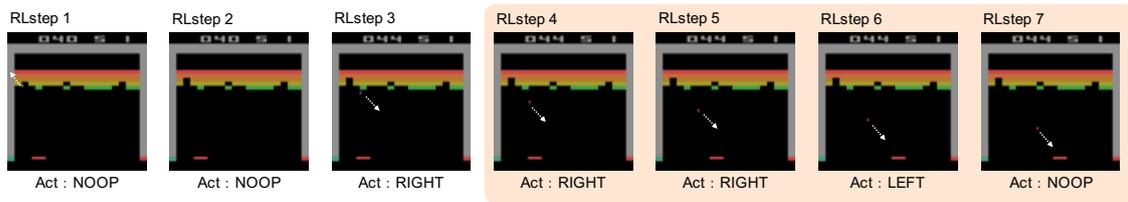
ChatGPT4oによる生成文（一部抜粋）

RLstep 78 : パドルを右に動かす (RIGHT)
 エージェントは画面中央を通過するボールを注視していますが、パドルの進行方向や位置には十分に注視していません。ボールへの注視を優先し、パドルの具体的な位置調整が遅れました。ボールはパドルを通過し、ミスとなりました。

(b) 長期説明プロンプト。上部は，重要イベント抽出にて抽出されたステップにおける視覚的説明情報とエージェントモデルの行動情報，下部は ChatGPT4o による生成文 (一部抜粋) を示し，オレンジ色でハイライトされたステップが該当する。

図 6.5: Breakout におけるボールの打ち返しに失敗したシーンに対する言語的説明例。ゲーム画面上の白矢印は，ボールの進行方向を示している。

weight を確認すると，全ステップにおいて一様に強く注視している領域がないことが分かる。そのため，RLstep 5 においてボールに対する注視は確認できず，注視対象に関する説明として不適切である。このように，制御に失敗していることを言語化できているが，注視対象に関する説明は適切ではないことを確認した。したがって，ChatGPT4o による言語的説明において，明確に注視領域が現れないシーンでの説明文生成は困難であると言える。図 6.5 (b) から長期説明プロンプトでは，視覚的説明情報として入力した動画データ等にもとづき，重要イベントとしてボールを打ち返すシーン (RLstep 39, 63, 206, 227) と，ボールの打ち返しに失敗したシーン (RLstep 78) を抽出している。ボールを打ち返せるかどうかは，Breakout においてスコアに直結する行動であることから，重要なイベントを抽出できていると考えられる。また，RLstep 78 において「画面中央を通過するボールを注視していますが，パドルの進行方向や位置には十分に注視していません」と述べ，「ボールへの注視を優先し，パドルの具体的な位置調整が遅れました」と制御に失敗していることを説明している。RLstep 78 の attention weight を確認すると，短期説明と同様に強く注視している領域が存在しないことが分かる。そのため，ボールへの注視に関する説明は不適切であると言える。このように長期説明においても，制御失敗については適切な説明が生成できているが，ボールへの注視については attention weight とは一致しない不適切な説明文を生成してしまっていることを確認した。

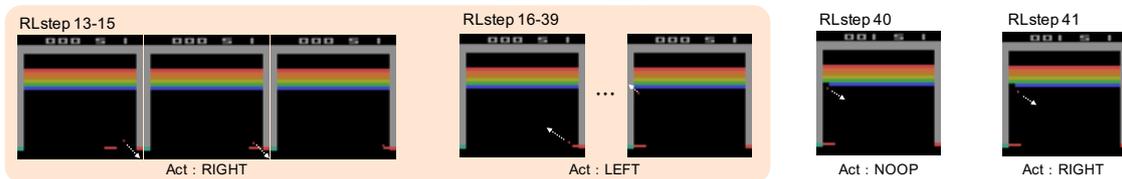


ChatGPT4oによる生成文（一部抜粋）

RLstep 4-7: ボールの跳ね返りとパドルの移動
 ボールが壁から跳ね返り、画面下方へ向かっている状態です。この段階で、エージェントは「RIGHT」を選択し、パドルを右に移動させています。その後、ボールの軌道に合わせて「LEFT」を選択して微調整し、パドルがボールの落下位置に合うように調整を行っています。ボールの軌道を追いながら、パドルがボールに確実に接触できるよう動かしたと考えられます。ボールが跳ね返る際の方向を予測し、パドルの位置を微調整しているのがエージェントの判断に対する根拠です。

(a) 短期説明プロンプト。上部は視覚的説明情報とエージェントモデルの行動情報、下部は ChatGPT4o による生成文 (一部抜粋) を示し、オレンジ色でハイライトされたステップが該当する。

重要イベント抽出



ChatGPT4oによる生成文（一部抜粋）

RLstep 13-15 (RIGHT)
 エージェントはボールの位置がパドルの右側に移動する可能性を予測し、パドルを右に動かす「RIGHT」の行動を選択したと考えられます。この間、ボールが画面内で適切にパドルに接触するよう動いている可能性が高いです。エージェントの選択は、次ステップでのパドルとボールの位置関係を注視した結果と考えられます。

RLstep 16-39 (LEFT)
 エージェントはRLstep13~15でボールを反射した後、ボールが画面内で左方向に移動していることを確認していると推測されます。このため、ボールを追尾する形でパドルを左に移動させる「LEFT」を選択しました。この区間では、パドルの移動がボールの動きに追従できているように見えます。しかし、連続で「LEFT」を選択し続ける間に、エージェントが動きすぎてボールを取り逃すリスクが生じる可能性もあります。

(b) 長期説明プロンプト。上部は、重要イベント抽出にて抽出されたステップにおける視覚的説明情報とエージェントモデルの行動情報、下部は ChatGPT4o による生成文 (一部抜粋) を示し、オレンジ色でハイライトされたステップが該当する。

図 6.6: Breakout におけるパドルにボールが向かうシーンに対する言語的説明例 (w/o attention weight)。ゲーム画面上の白矢印は、ボールの進行方向を示している。

■ attention weight の有無による生成文の比較

本項では、ChatGPT4o に提供する視覚的説明情報として、attention weight の有用性を検証する。視覚的説明情報としてゲーム画面 (RGB 画像) のみを用いる場合と、ゲーム画面 (RGB 画像) と attention weight を重畳させたマップを用いる場合で、ChatGPT4o が生成する説明文を比較する。比較シーンとして、図 6.4 で示した Breakout におけるボールがパドルに向かうシーンに焦点を当てる。図 6.6 に、視覚的説明情報としてゲーム画面 (RGB 画像) のみを用いた場合における言語的説明例を示す。

図 6.6 (a) から短期説明プロンプトでは、パドルの右側にボールが向かっている RLstep 4 - 7 において、「パドルがボールの落下位置に合うように調整を行っています」と生成している。これは、図 6.4 (a) の attention weight を用いた場合と同義な説明文である。一方で、根拠に対する説明は「パドルの

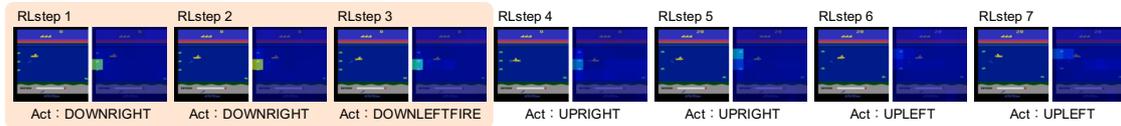
位置を微調整しているのがエージェントの判断に対する根拠です」とし、エージェントモデルの行動情報にもとづく説明文であることが分かる。また、長期説明プロンプトについては、図 6.6 (b) に示すように、重要イベントとしてボールがパドルに向かうシーン (RLstep 13-15) および、ボールを打ち返した後のシーン (RLstep 16-39) を抽出している。これは、attention weight を用いた場合と同様に、ボールの打ち返しに繋がる重要なイベントである。そのため、Breakout における重要イベントの抽出は attention weight を用いずとも可能であることを示唆している。一方で、エージェントの振る舞いに対する説明文は、「エージェントはボールの位置がパドルの右側に移動する可能性を予測し...」や「ボールが画面内で適切にパドルに接触するよう動いている可能性が高いです」など、ゲーム画面の遷移や行動情報をもとづいた予測による説明文であることが確認できる。したがって、attention weight はエージェントモデルが何に注目しているかという判断根拠となり、説明文の妥当性や信頼性を向上させる重要な情報であると考えられる。

6.2.2 Seaquest における言語的説明

Seaquest は、プレイヤーが自潜水艦 (黄色) を操作し、敵の潜水艦や魚を撃破または回避しつつダイバーを救出するゲームタスクである。エージェント (自潜水艦) には 18 つの行動がある: NOOP, FIRE, UP, DOWN, LEFT, RIGHT とこれらの組み合わせ。ここで、NOOP は動作なし、FIRE は自潜水艦のミサイル攻撃を意味する。Seaquest における言語的説明例を図 6.7 に示す。短期説明プロンプトは、異なる 2 つのシーン (敵である魚の撃破シーン、酸素を補給のために浮上するシーン) について示す。

図 6.7 (a) は、エージェントが魚 (敵) を撃破するシーンにおける短期説明プロンプトの言語的説明例である。この魚を撃破するシーンは、Seaquest においてスコア獲得に繋がる重要な行動である。図 6.7 (a) から短期説明プロンプトでは、RLstep 1,2,3 において「エージェントは、自潜水艦と画面左側の敵魚に注視しています」や「敵魚の位置に注視しています」など、注視対象が画面左の魚であると説明している。これは、RLstep 1,2,3 の attention weight ととも一致しているため、エージェントの注視領域を正しく捉えた説明文を生成できている。また、RLstep 3 では「敵魚が自機に近づいてくる状況を認識し、回避だけでなく攻撃行動 (FIRE) を実行しました」と、注視対象の魚に対する攻撃行動に言及している。RLstep 5 から、RLstep 3 における攻撃行動によって注視していた敵を撃破していることが確認でき、注視対象にもとづく攻撃行動を正しく説明できている。したがって、エージェントモデルによる敵の撃破行動について、注視対象を正確に述べた上で、エージェントモデルの振る舞いに対する説明できていると考えられる。

図 6.7 (b) は、エージェントが酸素を補給するために浮上するシーンにおける短期説明プロンプトの言語的説明例である。Seaquest において画面下部の酸素ゲージは、自潜水艦の残酸素量を表し、酸素量が空になるとゲーム終了となるため、海面に浮上し酸素を補給する行動が重要となる。図 6.7 (b) から短期説明プロンプトでは、RLstep 3 において「敵の位置を注視しています」や「右上に敵がいることを認識し...」など、注視対象が画面右上の敵であると説明している。これは、RLstep 3 の attention weight ととも一致しているため、エージェントの注視領域を正しく捉えた説明文を生成でき

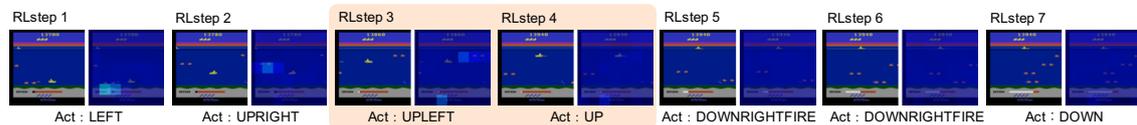


ChatGPT4oによる生成文（一部抜粋）

RLstep 1,2 (DOWNRIGHT)
 エージェントは、自潜水艦と画面左側の敵魚に注視しています。自機の安全を確保するため、敵魚が向かってくる進路から離れる行動（右下への移動）を選択しました。この行動は、敵との衝突を回避することを目的としていと考えられます。

RLstep 3 (DOWNLEFTFIRE)
 エージェントは、敵魚の位置に注視しています。敵魚が自機に近づいてくる状況を認識し、回避だけでなく攻撃行動（FIRE）を実行しました。この判断は、敵を撃破することで進路を確保しようとする意図を示しています。ただし、攻撃が成功しない場合、敵がさらに接近し危険な状況に陥るリスクがあります。

(a) 短期説明プロンプト: 敵である魚の撃破シーン。上部は視覚的説明情報とエージェントモデルの行動情報、下部は ChatGPT4o による生成文 (一部抜粋) を示し、オレンジ色でハイライトされたステップが該当する。



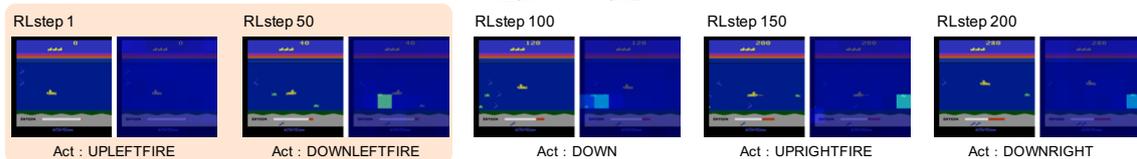
ChatGPT4oによる生成文（一部抜粋）

RLstep 3 (UPLEFT)
 エージェントは、潜水艦の進行方向と敵の位置に注視しています。右上に敵がいることを認識し、衝突のリスクが高まったため、エージェントは潜水艦を左上方向に移動 (UPLEFT) させました。この選択は、敵を避けつつ、安全な位置に進む意図がうかがえます。

RLstep 4 (UP)
 エージェントは、酸素ゲージに注視しています。酸素ゲージの残量が減少していることに認識し、酸素補給を目的として上昇する動き (UP) を選択しました。この行動は、酸素の枯渇を避けるために必要な判断でした。

(b) 短期説明プロンプト: 酸素を補給のために上浮するシーン。上部は視覚的説明情報とエージェントモデルの行動情報、下部は ChatGPT4o による生成文 (一部抜粋) を示し、オレンジ色でハイライトされたステップが該当する。

重要イベント抽出



ChatGPT4oによる生成文（一部抜粋）

RLstep 1 (UPLEFTFIRE)
 エージェントは、画面左上の敵潜水艦またはサメを注視していることがわかります。敵が画面左上に位置しているため、エージェントは上方向へ移動しながら攻撃 (UPLEFTFIRE) することで敵を排除しようとした。敵を無事に回避できている場合、この行動は適切だったといえます。一方で攻撃が当たらなかった場合、エージェントが敵との距離やタイミングを過小評価した可能性があります。

RLstep 50 (DOWNLEFTFIRE)
 エージェントは、画面左下にいる敵または障害物を注視している。敵や障害物との衝突を避けるため、左下方向へ移動しながら攻撃 (DOWNLEFTFIRE) する行動を選択。これは、敵との衝突を防ぎつつ攻撃を試みる意図があると考えられます。この行動が失敗した場合、敵が予想以上に高速で接近しており、攻撃よりも回避を優先するべきだった可能性があります。

(c) 長期説明プロンプト。上部は、重要イベント抽出にて抽出されたステップにおける視覚的説明情報とエージェントモデルの行動情報、下部は ChatGPT4o による生成文 (一部抜粋) を示し、オレンジ色でハイライトされたステップが該当する。

図 6.7: Seaquest における言語的説明例。短期説明プロンプトは、異なる 2 シーンでの言語的説明例を示す。

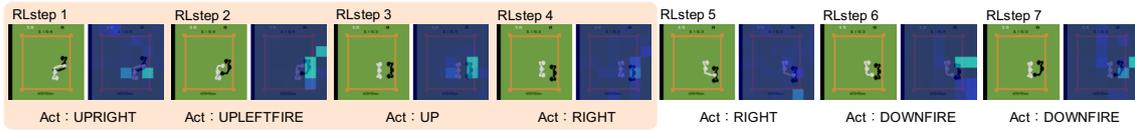
ている。さらに、RLstep 3 の行動 (UPLEFT) も注視対象の敵から遠ざかる行動であることから、「敵を避けつつ、安全な位置に進む意図がうかがえます」という説明も妥当であると考えられる。また、RLstep 4 において「酸素ゲージに注視しています」と注視対象が酸素ゲージであるとし、「酸素補給を目的として上昇する動き (UP) を選択しました」と酸素の補給行動について言及している。これは RLstep 4 の attention weight と一致し、RLstep 5 から酸素の補給に繋がっている。このことから、エージェントモデルによる酸素の補給行動に対しても、エージェントモデルの注視領域を正確に述べた上で、エージェントモデルの振る舞いに対する説明できていると考えられる。

図 6.7 (c) は、Seaquest における長期説明プロンプトによる言語的説明例である。図 6.7 (c) から長期説明プロンプトでは、視覚的説明情報として入力した動画データ等にもとづき、重要イベントとして敵を撃破するシーン (RLstep 50, 150) を抽出している。Seaquest において敵の撃破はスコア獲得に繋がる重要な行動であため、重要なイベントを抽出できていると考えられる。また、RLstep 50 において「画面左下にいる敵または障害物を注視している」と注視対象が敵であるとし、「敵との衝突を防ぎつつ攻撃を試みる意図があると考えられます」と敵との衝突を回避するために攻撃したと説明している。これは、短期説明プロンプト時 (図 6.7 (a) 参照) と同様の説明であり、敵の撃破行動について注視対象を含め、エージェントモデルの振る舞いに対する説明できている。一方で、RLstep 1 では「面左上の敵潜水艦またはサメを注視していることがわかります」と、敵を注視していると説明している。しかし、RLstep 1 の attention weight から、一様に強く注視している領域がないことが分かり、注視対象に関する説明として不適切であると言える。これは、Breakout における制御失敗したシーンに対する短期説明 (図 6.5 (a) 参照) と同様で、明確に注視領域が現れないシーンでの説明文生成は困難であったと考えられる。

6.2.3 Boxing における言語的説明

Boxing は、エージェントが白プレイヤーを操作して、敵の黒プレイヤーを倒すボクシングを模したゲームタスクである。エージェント (白プレイヤー) には 18 つの行動がある: NOOP, FIRE, UP, DOWN, LEFT, RIGHT とこれらの行動の組み合わせ。ここで FIRE は、白プレイヤーによるパンチを繰り出す行動である。Boxing における敵プレイヤーへ攻撃するシーンでの言語的説明例を図 6.8 に示す。Boxing では、敵の黒プレイヤーに攻撃を当てることでスコアを獲得できる。

図 6.8 (a) から短期説明プロンプトでは、RLstep 1, 2 において「距離を詰めながら攻撃を加えることで、より高得点を狙おうとしています」とエージェントの行動を説明している。RLstep 2 から攻撃行動 (FIRE) を選択していることが確認でき、Boxing では攻撃を当てるために敵へ近づく必要がある。したがって、エージェントの攻撃行動に対する説明として適していると考えられる。一方で、RLstep 1, 2 での「黒いボクサーの中心 (上半身付近) を注視しています」や「相手のガードが比較的下がっているように見え」、RLstep 3, 4 での「黒いボクサーの上半身を注視しています」など、敵の上半身に注視していると説明している。ここで、RLstep 1-4 の attention weight を確認すると、黒プレイヤーを注視していることが分かる。しかし、Boxing はプレイヤーを鳥瞰視点から見たようなゲームであり、上半身を注視しているかはマップから判断できないため、注視対象の説明として不適



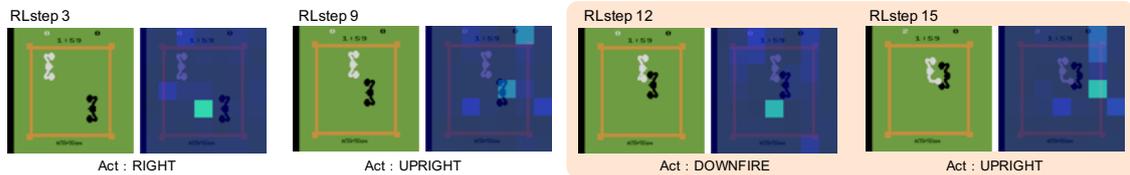
ChatGPT4oによる生成文（一部抜粋）

RLstep 1,2 (UPRIGHT, UPLEFTFIRE)
 エージェントは黒いボクサーの中心（上半身付近）を注視しています。このフレームで「UPRIGHT」や「UPELEFTFIRE」のような移動を伴う攻撃が選択されています。エージェントは相手の中心を狙い、距離を詰めながら攻撃を加えることで、より高得点を狙おうとしています。相手のガードが比較的下がっているように見え、これは頭部への攻撃が有効であると判断した可能性を示唆しています。

RLstep 3,4 (UP, RIGHT)
 エージェントは黒いボクサーの上半身を注視しています。この場面で「UP」や「RIGHT」のような移動操作が行われています。エージェントは相手との距離を調整しながら、攻撃が届く範囲に入り込むための位置取りを行っています。特にこの段階では相手が防御的な態勢をとっているため、攻撃を仕掛ける前に慎重に間合いを計っていると考えられます。

(a) 短期説明プロンプト。上部は視覚的説明情報とエージェントモデルの行動情報、下部は ChatGPT4o による生成文 (一部抜粋) を示し、オレンジ色でハイライトされたステップが該当する。

重要イベント抽出



ChatGPT4oによる生成文（一部抜粋）

RLstep 12 (DOWNFIRE)
 エージェントは画面中央下部を注視しており、相手キャラクターの体の下部（腹部または防御が薄い箇所）を注視しているように見えます。相手の防御が上半身に集中していることを認識し、下方からの攻撃が有効と判断して「DOWNFIRE」の行動を選択しました。この行動はスコアを稼ぐための戦略的な攻撃です。攻撃が相手に命中しましたが、大きなスコアには結びつきませんでした。これは、エージェントが近接距離を十分に保てていなかった可能性があります。

RLstep 15 (UPRIGHT)
 エージェントは画面中央上部を注視し、相手の顔付近や防御の隙間を見極めていると考えられます。エージェントは相手が防御を下に寄せた状態を確認し、上方向からの攻撃の有効性を判断して行動を変えました。この判断は、相手の動きに対応して得点を稼ぐ意図が伺えます。この行動は効果的であり、相手キャラクターを少し後退させることに成功しました。

(b) 長期説明プロンプト。上部は、重要イベント抽出にて抽出されたステップにおける視覚的説明情報とエージェントモデルの行動情報、下部は ChatGPT4o による生成文 (一部抜粋) を示し、オレンジ色でハイライトされたステップが該当する

図 6.8: Boxing における敵プレイヤーに攻撃するシーンに対する言語的説明例。

切であると言える。これは、ChatGPT4o が学習により保有する一般的な知識では、ゲーム画面が鳥瞰視点であることの認識が不足しているため、エージェントモデルの注視位置に対して誤った説明文を生成してしまったと考えられる。

6.2.4 言語的説明を活用したユーザに向けたマニュアル作成

深層強化学習エージェントモデルは、高い制御性能を獲得できることが知られており、ビデオゲーム攻略やロボティクスなど動的な制御が求められるタスクへ適用されている。前節では、それら深層強化学習エージェントモデルの意思決定に対する判断根拠の言語的説明について一部ビデオゲームを例に挙げ、ChatGPT4o により生成した説明文の妥当性を議論した。これらの言語的説明は、高性能な深層強化学習エージェントモデルの振る舞いに対する判断根拠をユーザが容易に理解可能な形で提供することができる。そこで本節では、前節で述べた深層強化学習エージェントの判断根拠

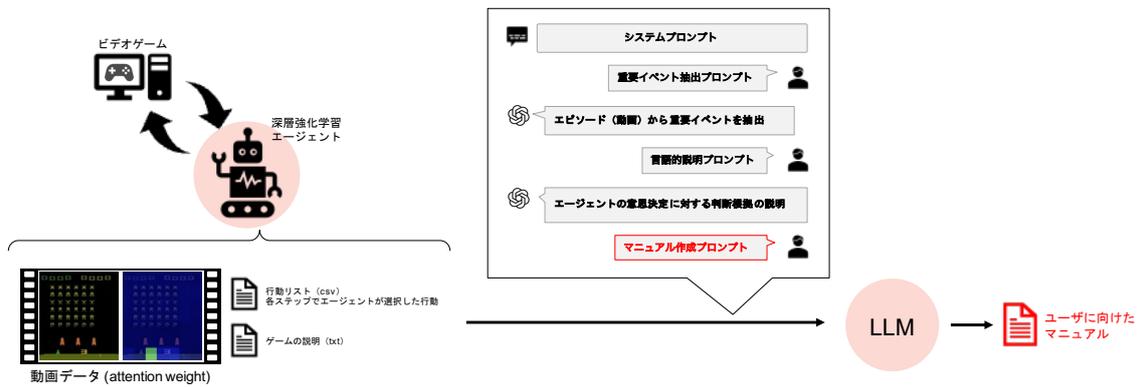


図 6.9: 言語的説明を活用したユーザーに向けたマニュアル作成の概略

に対する言語的説明の応用として、高性能な深層強化学習エージェントからユーザーがノウハウを学ぶためのマニュアル作成について述べる。

言語的説明を活用したマニュアル作成方法として、6.1.2 節で述べた長期説明プロンプトをベースとし、プロンプトにマニュアル作成プロンプトを追加することで、ChatGPT4o により言語的説明にもとづくマニュアル作成を行う。図 6.9 に言語的説明を活用したユーザーに向けたマニュアル作成の概略を示す。つまり、エージェントモデルの判断根拠に対する言語的説明までは、6.1.2 節で述べた長期説明プロンプトと同様である。プロンプトに記述したマニュアル作成のルールを以下に示す。

- エージェントモデルの判断根拠に対する言語的説明にもとづくマニュアルを作成するための指示。
例：エージェントの意思決定に対する判断根拠の説明から、特定のシーンにおけるユーザーの注視点（どこに注視したら良いか）と行動指針（どう制御すれば良いか）について言及してください。
- 参考フレームを明示するための指示。
例：ユーザーが注視点と行動指針を学ぶにあたり、何ステップあたりが参考になるかを補足してください。
- ユーザーが容易に理解できるようにするための指示。
例：誰にでもわかりやすい言葉で説明してください。

本節では、高性能なエージェントモデルからユーザーが学ぶためのマニュアル作成を目的とし、プレイヤーの戦略的な制御が求められる Space Invaders に焦点を当てる。Space Invaders は、画面上部から向かってくるインベーダーを撃破するゲームである。エージェント（大砲）には 6 つの行動がある：NOOP, FIRE, LEFT, RIGHT, LEFTFIRE, RIGHTFIRE。ここで、FIRE は大砲による射撃行動を意味する。Space Invaders における言語的説明を活用したマニュアル作成例を図 6.10 に示す。図 6.10 から、重要イベント抽出段階において、敵の接近、宇宙船の出現、シールドの破壊、残存の敵に関するイベントを抽出している。Space Invaders は、インベーダーが画面下部に到達することと、エー



図 6.10: Space Invaders における言語的説明を活用したマニュアル作成例。

ジェントがインベーダーの攻撃に当たることでゲームが終了する。また、プレイ途中に出現する宇宙船は撃破すると大きなスコア獲得に繋がり、エージェント上部のシールドはインベーダーの攻撃を防ぐことが可能である。したがって、抽出したイベントは、Space Invaders における制御において重要なイベントを正しく抽出できている。言語的説明段階では、「注視した敵が地面に接近する危険を認識し、射撃で早急に排除する判断を下した」（敵の接近）や「敵の爆撃によりシールドが破壊されつつある状況を確認し、そのエリアに近い敵を排除しようと試みました」（シールドの破壊）など、注視対象にもとづきエージェントの行動を説明している。イベントごとの attention weight から、一貫してエージェント周辺を強く注視しているが、イベントごとに特徴的なオブジェクト（敵が接近であれば最も画面下部に近い敵、残りの敵が 1 体では最後に残った敵周辺、etc.）を注視していることがわかる。このように各イベントに対し、注視対象という根拠を踏まえつつ、エージェントモデルの振る舞いに対する説明文を生成できていることが確認できる。そして、マニュアル作成段階では、「画面中央下部で、敵が地面に近づくエリアに注目してください。特に、接近速度が速い敵は優先的に排除すべきです」（敵の接近）と述べ、言語的説明時の説明に準拠したアドバイスを生成している。他のイベントも同様に、「敵の爆撃でシールドが破壊されつつあるエリアに注目します」や「シールドを攻撃する敵を優先的に排除しつつ、移動しながら射撃を行ってシールド周辺をクリアにしてください」（シールドの破壊）と述べ、近い敵の排除など言語的説明にもとづき、ユーザへのアドバイスを生成している。このように、ユーザがプレイする上でどこを注視すれば良いか、またどのような行動指針で制御すれば良いかを、言語的説明段階で述べた点に着目しつつ、ユーザへのアドバイスを提供できることを確認した。

6.3 まとめ

本章では、深層強化学習エージェントモデルの意思決定プロセスに対する判断根拠の説明性/解釈性向上を目的とし、4, 5章で述べたような視覚的説明手法から、ユーザが容易に解釈できる自然言語を用いた言語的説明への拡張について議論した。本研究では、深層強化学習エージェントに対する視覚的説明情報を活用した大規模言語モデル (LLM) による言語的説明手法を提案した。Atari2600における3つのビデオゲーム (Breakout, Seaquest, Boxing) に焦点を当て、本手法の有用性について検証した。評価実験では、LLMとしてChatGPT4oを採用し、視覚的説明情報の取得には5章で提案したAction Q-Transformer (AQT)を用いた。これらの実験から、BreakoutやSeaquestにおけるいくつかのシーンにおいて、エージェントモデルの注視対象にもとづき、エージェントモデルの振る舞いに対する解釈性の高い説明文を生成できることを確認した。一方で、エージェントモデルが制御を失敗したシーンなどのエージェントモデルの注視領域が明確に現れないシーンでは、注視対象に対する誤った説明が見られた。また、BoxingのようなChatGPT4oが持つ知識と乖離があるタスクでは、説明文生成が難しいタスクでは、ChatGPT4oが誤った説明文を生成してしまうことを確認した。本手法におけるChatGPT4oに対する外部知識は、Domain knowledge databaseに獲得したドメイン知識情報をもとに提供する。このドメイン知識情報は、それぞれのビデオゲームタスクの概要やプレイ方法など基礎的な情報が主であり、ゲームの攻略方法等は含まれていない。そのため、ChatGPT4oが保有する知識では不十分であるケースが発生し、誤った説明文が生成されたと考えられる。今後の課題として、明確に注視領域が現れないシーンに対して、注視していないと説明できるようにLLMをInstruction Tuningを適用することや、重要イベント抽出に特化したLLMの開発、人が持つ知見 (ビデオゲームにおける攻略法など) をDomain knowledge databaseに蓄積することで外部知識の質を向上、生成された説明文に対する定量的評価などが挙げられる。

6.2.4節では、これらの言語的説明によって得られるエージェントモデルの判断根拠に対する説明文の活用方法として、言語的説明を活用したユーザに向けたマニュアル作成を示した。プレイヤーの戦略的な制御に求められるSpace Invadersに焦点を当て、高性能な深層強化学習エージェントモデルから、Space Invadersをユーザがプレイする上で注視ポイントや行動指針など、言語的説明段階で述べた点に着目しつつ攻略マニュアルを生成できることを示した。本研究では、ビデオゲームタスクのみでの実証実験である。そのため、自動運転などビデオゲーム以外の実用的なタスクでの実証実験への取り組みが、今後の課題として挙げられる。

第7章

結論と展望

本稿では、深層強化学習の代表的な課題である学習効率の低さとエージェントモデルの信頼性に焦点を当てた。学習効率の低さの解決に向け、補助学習に対する補助タスクの動的な選択に関するアプローチを提案した。エージェントモデルのブラックボックス問題の解決に向け、エージェントモデルの方策と状態価値に着目した視覚的説明手法や、エージェントモデルが選択可能な行動全てに着目した視覚的説明手法を提案した。また、ユーザに対するエージェントモデルの説明性/解釈性向上にむけ、視覚的説明から自然言語による言語的説明への拡張について議論した。以下に、本論文の結論と今後の展望について述べる。

7.1 結論

各章のまとめは以下の通りである。2章では、方策の学習方法による分類に焦点を当て、深層強化学習アルゴリズムを体系的にまとめたあと、その応用事例と課題について述べた。また、深層強化学習の代表的な課題として、学習効率の低さとエージェントモデルの信頼性に着目し、説明可能な強化学習についての関連研究をまとめた。説明可能な強化学習には、エージェントモデルの意思決定に対する判断根拠を顕著性マップや attention map などの画像情報として提供する視覚的説明、自然言語によるテキスト情報として提供する言語的説明にカテゴリ化できる。前者は、摂動を用いるアプローチと、attention を用いるアプローチに分類し、それぞれの代表的ないくつかの手法について説明した。後者は、テンプレート文を用いた説明文生成アプローチと、自由形式での説明文生成アプローチに分類し、それぞれの代表的ないくつかの手法について説明したのち、エージェントモデルに対する言語的説明のメリットについて述べた。

3章では、学習効率の低さに対処した補助タスクを用いる深層強化学習手法に焦点を当て、エージェントモデルの学習対象であるメインタスクに応じて補助タスクを動的に選択するモジュール Auxiliary Selection (AS) を提案した。ビデオゲーム攻略タスクにおいて教師なし補助タスクを用いる UNREAL に AS を適用し、補助タスクを動的に選択できることを検証した。この結果から、メインタスクに対して有効な補助タスクの人手による選定が必要ないことを示し、教師なし補助タスクのような有効か判断が困難な補助タスクにおいても、メインタスクの学習効率向上を確認した。また、AS はメインタスクの学習段階に応じた最適な補助タスクの選択を可能とし、メインタスクの学習状況に応じた性能向上を実現した。

4章では、深層強化学習エージェントの方策と状態価値に着目し、Actor-Critic 法にもとづく深層強

化学習エージェントの判断根拠に対する視覚的説明手法 Mask attention A3C (Mask A3C) を提案した。Mask A3C は、Actor-Critic 法の output branch に attention 機構を導入することで、出力値である方策と状態価値のそれぞれに関連した重要な領域を示す mask-attention を生成する。この mask-attention を可視化することで、エージェントモデルの意思決定プロセスに対する網羅的な洞察が視覚的に得られることを確認した。

5章では、行動情報を query として用いる Transformer encoder-decoder 構造にもとづく深層強化学習エージェントの視覚的説明手法 Action Q-Transformer (AQT) を提案した。AQT における decoder 部では、入力画像に対する特徴マップと行動情報を表現した action query 間での cross-attention により、行動ごとに固有の attention weight を獲得する。これにより、エージェントが選択可能な行動全てに対する attention weight を生成でき、エージェントモデルの意思決定プロセスに対する網羅的かつ視覚的な洞察を提供可能とした。また、ロボット制御のような実空間上で動作するタスクにおいて、ユーザがエージェントモデルの判断根拠を直感的かつ即時的に理解できる仕組みとして、ユーザへの Augmented Reality (AR) を用いた視覚的フィードバック手法を提案した。ロボットエージェントの行動選択に対する attention weight を、実空間上で AR を用いて可視化することで、エージェントモデルの判断根拠に対するユーザへの効果的な説明の提供を実現した。

6章では、4章と5章で提案したような視覚的説明手法から、エージェントモデルの判断根拠に対する言語的説明への拡張について議論した。深層強化学習エージェントに対する視覚的説明情報を活用した大規模言語モデル (LLM) による言語的説明手法を示し、Atari2600 の一部ビデオゲームに焦点を当て、ChatGPT4o を用いた言語的説明例を示した。これらの言語的説明例から、妥当な説明文を生成できた事例とできなかった事例をまとめ、エージェントモデルの断根拠を示す解釈性の高い言語的説明に向けた課題について議論した。また、エージェントモデルの判断根拠に対する言語的説明を活用した応用例として、高性能な深層強化学習エージェントからユーザが学ぶためのマニュアル作成について挙げ、さらなる発展に向け議論した。

7.2 展望

本論文では、深層強化学習におけるエージェントモデルの性能向上と意思決定プロセスに対する説明/解釈性向上に関する研究を行った。深層強化学習には、学習効率の低さに起因するエージェントモデルの性能に対する課題と、高性能なエージェントモデルの意思決定プロセスがブラックボックスであることに起因するエージェントモデルの信頼性に対する課題がある。

エージェントモデルの性能に対する課題の解決策として、エージェントモデルの学習時において、メインタスクとは異なる目的を持った補助タスクを導入によるメインタスクの学習効率向上が挙げられる。しかしながら、これらの補助タスクはメインタスクにとって必ずしも有効であるとは限らない。本論文は、この課題に対して、メインタスクに適した動的な補助タスクの選択アプローチを提案した。教師なし補助タスクを用いたビデオゲームタスクにおける実験により、メインタスクの学習段階に合わせた補助タスクの選択を確認した。本研究では、ビデオゲームタスクのみでの有効性を検証していないため、自動運転タスクなど複雑かつ実応用に近いタスクにおける実験が今後の研

究課題である。

エージェントモデルの信頼性に対する課題は、深層強化学習を利用したシステムの信頼性に直結するため、深層強化学習の応用に対し大きな障害となる。本論文は、この課題に対してエージェントモデルの判断根拠に対する視覚的説明のアプローチを提案し、それら視覚的説明情報にもとづく LLM による言語的説明への拡張方法について提案した。これらの視覚的説明と言語的説明は、医療や自動運転、ロボティクス、金融などセンシティブなタスクに対する深層強化学習の応用に有用であると考えられる。例えば、医療分野では治療法の推薦システムに対する推薦根拠の生成や、自動運転分野では制御システムに対する制御理由の生成が挙げられる。これらシステムの利用者は、AI に関して豊富な知見を有した専門家ではないため、ユーザに対して分かりやすく説明を提供できる高い解釈性を有した深層強化学習エージェントモデルの確立が必要であると考え。ここで、ユーザに対する高い解釈性を有するアプローチとして、ユーザの主観に依存しにくい自然言語による言語的説明が適している。本論文では、視覚的説明情報にもとづく LLM による言語的説明への拡張について述べたが、解釈可能な説明を生成できるシーンは限定的である。したがって、深層強化学習エージェントモデルの意思決定プロセスに対して、ユーザが容易に理解可能な解釈性の高い自然言語による言語的説明の確立が今後の研究課題である。また、これらの言語的説明の技術が確立したのちには、高性能な深層強化学習エージェントモデルが持つ知見を、ユーザへ教示できるようなソリューションの実現が期待できる。

謝 辞

本研究の遂行にあたり，常日頃ご指導を賜りました中部大学 工学部情報工学科 山下隆義 教授に深く感謝の意を表します．本論文をまとめるにあたり，有益なご討論，ご助言を賜りました中部大学 工学部情報工学科 山内康一郎 教授，中部大学 理工学部 AI ロボティクス学科 藤吉弘巨 教授，慶應義塾大学 理工学部情報工学科 杉浦孔明 教授に謹んで感謝いたします．本研究において，貴重なご意見，ご指導を頂きました中部大学 理工学部 AI ロボティクス学科 藤吉弘巨 教授，中部大学 AI 数理データサイエンスセンター 平川翼 講師，慶應義塾大学 理工学部情報工学科 杉浦孔明 教授に心から厚く御礼申し上げます．最後に，本研究にご協力して頂いた山下研究室と藤吉研究室の皆様に感謝致します．

参考文献

- [1] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, “Rainbow: Combining improvements in deep reinforcement learning”, Association for the Advancement of Artificial Intelligence, 2018.
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need”, Advances in Neural Information Processing Systems, vol.30, 2017.
- [3] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mor-datch, “Decision transformer: Reinforcement learning via sequence modeling”, Advances in Neural Information Processing Systems, vol.34, pp.15084–15097, 2021.
- [4] S. Greydanus, A. Koul, J. Dodge, and A. Fern, “Visualizing and understanding Atari agents”, International Conference on Machine Learning, vol.80, pp.1792–1801, 2018.
- [5] A. Mott, D. Zoran, M. Chrzanowski, D. Wierstra, and D. Jimenez Rezende, “Towards interpretable reinforcement learning using attention augmented agents”, Advances in Neural Information Process-ing Systems, vol.32, 2019.
- [6] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., “Mastering the game of Go with deep neural networks and tree search”, Nature, vol.529, no.7587, pp.484–489, 2016.
- [7] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Ku-maran, T. Graepel, et al., “A general reinforcement learning algorithm that masters chess, shogi, and go through self-play”, Science, vol.362, no.6419, pp.1140–1144, 2018.
- [8] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrish-nan, V. Vanhoucke, et al., “Scalable deep reinforcement learning for vision-based robotic manipu-lation”, Conference on Robot Learning, pp.651–673, 2018.

- [9] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J. M. Allen, V. D. Lam, A. Bewley, and A. Shah, “Learning to drive in a day”, International Conference on Robotics and Automation, pp.8248-8254, 2018.
- [10] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., “Training language models to follow instructions with human feedback”, Advances in Neural Information Processing Systems, vol.35, pp.27730–27744, 2022.
- [11] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine, “Diversity is all you need: Learning skills without a reward function”, International Conference on Learning Representations, 2019.
- [12] T. Schaul, D. Horgan, K. Gregor, and D. Silver, “Universal value function approximators”, International Conference on Machine Learning, vol.37, pp.1312–1320, 2015.
- [13] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu, “Reinforcement learning with unsupervised auxiliary tasks”, International Conference on Learning Representations, 2017.
- [14] B. Hayes, and J. A. Shah, “Improving robot controller transparency through autonomous policy explanation”, ACM/IEEE international conference on human-robot interaction, pp.303–312, 2017.
- [15] U. Ehsan, B. Harrison, L. Chan, and M. O. Riedl, “Rationalization: A neural machine translation approach to generating natural language explanations”, AAAI/ACM Conference on AI, Ethics, and Society, p.81–87, 2018.
- [16] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding”, North American Chapter of the Association for Computational Linguistics, 2019.
- [17] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training”, OpenAI Blog, 2018.
- [18] T. B. Brown, “Language models are few-shot learners”, *arXiv preprint arXiv:2005.14165*, 2020.
- [19] R. S. Sutton, and A. G. Barto, Reinforcement learning: An introduction, MIT press, 1988.
- [20] V. Firoiu, W. F. Whitney, and J. B. Tenenbaum, “Beating the world’s best at super smash bros. melee with deep reinforcement learning”, *arXiv preprint, arXiv:1702.06230*, 2017.
- [21] N. Justesen, P. Bontrager, J. Togelius, and S. Risi, “Deep learning for video game playing”, *arXiv preprint, arXiv:1708.07902*, 2017.

- [22] S. Gu, E. Holly, T. Lillicrap, and S. Levine, “Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates”, *International Conference on Robotics and Automation*, pp.3389–3396, 2017.
- [23] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey”, *The International Journal of Robotics Research*, vol.32, no.11, pp.1238–1274, 2013.
- [24] C. J. Watkins, and P. Dayan, “Q-learning”, *Machine learning*, vol.8, no.3-4, pp.279-292, 1992.
- [25] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., “Human-level control through deep reinforcement learning”, *Nature*, vol.518, no.7540, pp.529-533, 2015.
- [26] H. Van Hasselt, A. Guez, and D. Silver, “Deep Reinforcement Learning with Double Q-Learning”, *Association for the Advancement of Artificial Intelligence*, 2016.
- [27] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, “Dueling network architectures for deep reinforcement learning”, *International Conference on Machine Learning*, pp.1995–2003, 2016.
- [28] M. Fortunato, M. G. Azar, B. Piot, J. Menick, M. Hessel, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin, C. Blundell, and S. Legg, “Noisy networks for exploration”, *International Conference on Learning Representations*, 2018.
- [29] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, “Prioritized experience replay”, *arXiv preprint arXiv:1511.05952*, 2015.
- [30] M. G. Bellemare, W. Dabney, and R. Munos, “A distributional perspective on reinforcement learning”, *International Conference on Machine Learning*, pp.449–548, 2017.
- [31] A. Nair, P. Srinivasan, S. Blackwell, C. Alcicek, R. Fearon, A. De Maria, V. Panneershelvam, M. Suleyman, C. Beattie, S. Petersen, et al., “Massively parallel methods for deep reinforcement learning”, *International Conference on Machine Learning deep learning workshop*, 2015.
- [32] D. Horgan, J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. Van Hasselt, and D. Silver, “Distributed prioritized experience replay”, *International Conference on Learning Representations*, 2018.
- [33] S. Kapturowski, G. Ostrovski, J. Quan, R. Munos, and W. Dabney, “Recurrent experience replay in distributed reinforcement learning”, *International Conference on Learning Representations*, 2019.
- [34] A. Puigdomènech Badia, P. Sprechmann, A. Vitvitskyi, D. Guo, B. Piot, S. Kapturowski, O. Tieleman, M. Arjovsky, A. Pritzel, A. Bolt, and C. Blundell, “Never give up: Learning directed exploration strategies”, *International Conference on Learning Representations*, 2020.

- [35] T. L. Paine, C. Gulcehre, B. Shahriari, M. Denil, M. Hoffman, H. Soyer, R. Tanburn, S. Kapturowski, N. Rabinowitz, D. Williams, et al., “Making Efficient Use of Demonstrations to Solve Hard Exploration Problems”, International Conference on Learning Representations, 2020.
- [36] A. P. Badia, B. Piot, S. Kapturowski, P. Sprechmann, A. Vitvitskyi, D. Guo, and C. Blundell, “Agent57: Outperforming the atari human benchmark”, International Conference on Machine Learning, 2020.
- [37] S. Kapturowski, V. Campos, R. Jiang, N. Rakićević, H. van Hasselt, C. Blundell, and A. P. Badia, “Human-level atari 200x faster”, *arXiv preprint arXiv:2209.07550*, 2022.
- [38] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning”, *Machine learning*, vol.8, no.3, pp.229–256, 1992.
- [39] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation”, *Advances in Neural Information Processing Systems*, vol.12, pp.1057–1063, 1999.
- [40] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, “Trust region policy optimization”, International Conference on Machine Learning, pp.1889–1897, 2015.
- [41] J. d. F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms”, *arXiv preprint arXiv:1707.06347*, 2017.
- [42] V. R. Konda, and J. N. Tsitsiklis, “Actor-critic algorithms”, *Advances in Neural Information Processing Systems*, pp.1008–1014, 2000.
- [43] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning”, International Conference on Learning Representations, 2016.
- [44] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning”, International Conference on Machine Learning, pp.1928–1937, 2016.
- [45] Y. Wu, E. Mansimov, R. B. Grosse, S. Liao, and J. Ba, “Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation”, *Advances in Neural Information Processing Systems*, vol.30, 2017.
- [46] Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, and N. de Freitas, “Sample efficient actor-critic with experience replay”, International Conference on Learning Representations, 2017.

- [47] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, et al., “Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures”, *International Conference on Machine Learning*, pp.1407–1416, 2018.
- [48] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor”, *International Conference on Machine Learning*, 2018.
- [49] G. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, D. TB, A. Muldal, N. Heess, and T. Lillicrap, “Distributional policy gradients”, *International Conference on Learning Representations*, 2018.
- [50] S. Fujimoto, H. Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods”, *International Conference on Machine Learning*, pp.1587–1596, 2018.
- [51] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale”, *International Conference on Learning Representations*, 2021.
- [52] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers”, *European Conference on Computer Vision*, pp.213–229, 2020.
- [53] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention”, *International Conference on Machine Learning*, pp.10347–10357, 2021.
- [54] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations”, *Advances in Neural Information Processing Systems*, vol.33, pp.12449–12460, 2020.
- [55] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, “Vivit: A video vision transformer”, *International Conference on Computer Vision*, pp.6836–6846, 2021.
- [56] K. Fang, A. Toshev, L. Fei-Fei, and S. Savarese, “Scene memory transformer for embodied agents in long-horizon tasks”, *Computer Vision and Pattern Recognition*, pp.538–547, 2019.
- [57] M. Janner, Q. Li, and S. Levine, “Offline reinforcement learning as one big sequence modeling problem”, *Advances in Neural Information Processing Systems*, vol.34, pp.1273–1286, 2021.
- [58] K. H. Lee, O. Nachum, M. Yang, L. Y. Lee, D. Freeman, W. Xu, S. Guadarrama, I. S. Fischer, E. Jang, H. Michalewski, and I. Mordatch, “Multi-game decision transformers”, *Advances in Neural Information Processing Systems*, 2022.

- [59] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-maroon, M. Giménez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas, “A generalist agent”, *Transactions on Machine Learning Research*, 2022.
- [60] Q. Zheng, A. Zhang, and A. Grover, “Online decision transformer”, *International Conference on Machine Learning*, pp.27042–27059, 2022.
- [61] T. Yamagata, A. Khalil, and R. Santos-Rodriguez, “Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline rl”, *arXiv preprint arXiv:2209.03993*, 2022.
- [62] J. Shang, K. Kahatapitiya, X. Li, and M. S. Ryoo, “Starformer: Transformer with state-action-reward representations for visual reinforcement learning”, *European Conference on Computer Vision*, pp.462–479, 2022.
- [63] Y. Chebotar, Q. Vuong, K. Hausman, F. Xia, Y. Lu, A. Irpan, A. Kumar, T. Yu, A. Herzog, K. Pertsch, et al., “Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions”, *Conference on Robot Learning*, pp.3909–3928, 2023.
- [64] P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-attention with relative position representations”, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp.464–468, Association for Computational Linguistics, 2018.
- [65] M. A. Islam, S. Jia, and N. D. B. Bruce, “How much position information do convolutional neural networks encode?”, *International Conference on Learning Representations*, 2020.
- [66] J. Kaplan, S. McCandlish, T. J. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, “Scaling laws for neural language models”, *arXiv preprint, arXiv:2001.08361*, 2020.
- [67] T. Henighan, J. Kaplan, M. Katz, M. Chen, C. Hesse, J. Jackson, H. Jun, T. B. Brown, P. Dhariwal, S. Gray, et al., “Scaling laws for autoregressive generative modeling”, *arXiv preprint arXiv:2010.14701*, 2020.
- [68] J. T. Springenberg, A. Abdolmaleki, J. Zhang, O. Groth, M. Bloesch, T. Lampe, P. Brakel, S. M. E. Bechtle, S. Kapturovski, R. Hafner, N. Heess, and M. Riedmiller, “Offline actor-critic reinforcement learning scales to large models”, *International Conference on Machine Learning*, 2024.

- [69] M. Laskin, L. Wang, J. Oh, E. Parisotto, S. Spencer, R. Steigerwald, D. Strouse, S. S. Hansen, A. Filos, E. Brooks, maxime gazeau, H. Sahni, S. Singh, and V. Mnih, “In-context reinforcement learning with algorithm distillation”, International Conference on Learning Representations, 2023.
- [70] Łukasz Kaiser, M. Babaeizadeh, P. Miłoś, B. Osiński, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, A. Mohiuddin, R. Sepassi, G. Tucker, and H. Michalewski, “Model based reinforcement learning for atari”, International Conference on Learning Representations, 2020.
- [71] P. R. Wurman, S. Barrett, K. Kawamoto, J. MacGlashan, K. Subramanian, T. J. Walsh, R. Capobianco, A. Devlic, F. Eckert, F. Fuchs, et al., “Outracing champion gran turismo drivers with deep reinforcement learning”, *Nature*, vol.602, no.7896, pp.223–228, 2022.
- [72] Y. F. Chen, M. Everett, M. Liu, and J. P. How, “Socially aware motion planning with deep reinforcement learning”, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1343–1350, 2017.
- [73] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control”, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.5026–5033, 2012.
- [74] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, et al., “Isaac gym: High performance gpu-based physics simulation for robot learning”, Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), 2021.
- [75] Z. Qiu, S. Chen, J. Shi, F. Wang, and N. Zheng, “Human-like reverse parking using deep reinforcement learning with attention mechanism”, IEEE Intelligent Vehicles Symposium, pp.2553–2560, 2024.
- [76] P. Wang, and C. Y. Chan, “Formulation of deep reinforcement learning architecture toward autonomous driving for on-ramp merge”, International Conference on Intelligent Transportation Systems, pp.1–6, 2017.
- [77] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, “Navigating occluded intersections with autonomous vehicles using deep reinforcement learning”, International Conference on Robotics and Automation, pp.2034–2039, 2018.
- [78] A. Coronato, M. Naeem, G. D. Pietro, and G. Paragliola, “Reinforcement learning for intelligent healthcare applications: A survey”, *Artificial intelligence in medicine*, vol.109, p.101964, 2020.

- [79] S. Gönül, T. Namlı, A. Coşar, and İ. H. Toroslu, “A reinforcement learning based algorithm for personalization of digital, just-in-time, adaptive interventions”, *Artificial Intelligence in Medicine*, vol.115, p.102062, 2021.
- [80] E. Benhamou, D. Saltiel, J. J. Ohana, and J. Atif, “Detecting and adapting to crisis pattern with context based deep reinforcement learning”, *International Conference on Pattern Recognition*, pp.10050-10057, 2020.
- [81] X. Y. Liu, H. Yang, J. Gao, and C. D. Wang, “Finrl: Deep reinforcement learning framework to automate trading in quantitative finance”, *Proceedings of the second ACM international conference on AI in finance*, pp.1–9, 2021.
- [82] J. Degraeve, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de Las Casas, et al., “Magnetic control of tokamak plasmas through deep reinforcement learning”, *Nature*, vol.602, no.7897, pp.414–419, 2022.
- [83] P. Razzaghi, A. Tabrizian, W. Guo, S. Chen, A. Taye, E. E. Thompson, A. Bregeon, A. Baheri, and P. Wei, “A survey on reinforcement learning in aviation applications”, *Engineering Applications of Artificial Intelligence*, vol.136, p.108911, 2022.
- [84] G. Waxenegger-Wilfing, K. Dresia, J. Deeken, and M. Oswald, “A reinforcement learning approach for transient control of liquid rocket engines”, *IEEE Transactions on Aerospace and Electronic Systems*, vol.57, no.5, pp.2938–2952, 2021.
- [85] W. Li, Y. Zhu, and D. Zhao, “Missile guidance with assisted deep reinforcement learning for head-on interception of maneuvering target”, *Complex & Intelligent Systems*, vol.8, no.2, pp.1205–1216, 2022.
- [86] X. Chen, L. Yao, J. McAuley, G. Zhou, and X. Wang, “Deep reinforcement learning in recommender systems: A survey and new perspectives”, *Knowledge-Based Systems*, vol.264, p.110335, 2023.
- [87] J. Luo, C. Paduraru, O. Voicu, Y. Chervonyi, S. A. Munns, J. Li, C. Qian, P. Dutta, J. Q. Davis, N. Wu, X. Yang, C. M. Chang, T. Li, R. Rose, M. Fan, H. Nakhost, T. Liu, B. Kirkman, F. Altamura, L. Cline, P. Tonker, J. P. Gouker, D. Udén, W. B. Bryan, J. Law, D. Fatiha, N. Satra, J. Rothenberg, M. A. Carlin, S. Tallapaka, S. Witherspoon, D. Parish, P. Dolan, C. Zhao, and D. J. Mankowitz, “Controlling commercial cooling systems using reinforcement learning”, *arXiv preprint arXiv:2211.07357*, 2022.
- [88] G. Ostrovski, M. G. Bellemare, A. Oord, and R. Munos, “Count-based exploration with neural density models”, *International Conference on Machine Learning*, pp.2721–2730, 2017.

- [89] R. Houthoofd, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel, “Vime: Variational information maximizing exploration”, *Advances in Neural Information Processing Systems*, vol.29, 2016.
- [90] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, “Curiosity-driven exploration by self-supervised prediction”, *International Conference on Machine Learning*, 2017.
- [91] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, “Exploration by random network distillation”, *International Conference on Learning Representations*, 2019.
- [92] A. Y. Ng, and S. Russell, “Algorithms for inverse reinforcement learning”, *International Conference on Machine Learning*, 2000.
- [93] P. Abbeel, and A. Y. Ng, “Apprenticeship learning via inverse reinforcement learning”, *International Conference on Machine Learning*, p.1, 2004.
- [94] D. Ramachandran, and E. Amir, “Bayesian inverse reinforcement learning.”, *International Joint Conferences on Artificial Intelligence*, vol.7, pp.2586–2591, 2007.
- [95] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, et al., “Maximum entropy inverse reinforcement learning.”, *Association for the Advancement of Artificial Intelligence*, vol.8, pp.1433–1438, 2008.
- [96] J. Fu, K. Luo, and S. Levine, “Learning robust rewards with adversarial inverse reinforcement learning”, *International Conference on Learning Representations*, 2018.
- [97] J. Choi, and K. E. Kim, “Inverse reinforcement learning in partially observable environments”, *Journal of Machine Learning Research*, vol.12, pp.691-730, 2009.
- [98] O. Nachum, S. S. Gu, H. Lee, and S. Levine, “Data-efficient hierarchical reinforcement learning”, *Advances in Neural Information Processing Systems*, vol.31, 2018.
- [99] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra, “Pathnet: Evolution channels gradient descent in super neural networks”, *arXiv preprint arXiv:1701.08734*, 2017.
- [100] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, “Progressive neural networks”, *arXiv preprint arXiv:1606.04671*, 2016.
- [101] T. Wang, D. Lizotte, M. Bowling, and D. Schuurmans, “Bayesian sparse sampling for on-line reward optimization”, *International Conference on Machine Learning*, pp.956–963, 2005.
- [102] D. Silver, and J. Veness, “Monte-carlo planning in large pomdps”, *Advances in Neural Information Processing Systems*, vol.23, 2010.

- [103] Y. Wang, K. S. Won, D. Hsu, and W. S. Lee, “Monte carlo bayesian reinforcement learning”, International Conference on Machine Learning, 2012.
- [104] M. Hong, Z. Qi, and Y. Xu, “Model-based reinforcement learning for confounded POMDPs”, International Conference on Machine Learning, vol.235, pp.18668–18710, 2024.
- [105] D. Ha, and J. Schmidhuber, “World models”, *arXiv preprint arXiv:1803.10122*, 2018.
- [106] D. Hafner, T. P. Lillicrap, M. Norouzi, and J. Ba, “Mastering atari with discrete world models”, International Conference on Learning Representations, 2021.
- [107] V. Micheli, E. Alonso, and F. Fleuret, “Transformers are sample-efficient world models”, International Conference on Learning Representations, 2023.
- [108] C. Finn, T. Yu, J. Fu, P. Abbeel, and S. Levine, “Generalizing skills with semi-supervised reinforcement learning”, International Conference on Learning Representations, 2017.
- [109] Z. Xu, H. P. van Hasselt, and D. Silver, “Meta-gradient reinforcement learning”, *Advances in Neural Information Processing Systems*, vol.31, 2018.
- [110] R. Mendonca, A. Gupta, R. Kravev, P. Abbeel, S. Levine, and C. Finn, “Guided meta-policy search”, *Advances in Neural Information Processing Systems*, vol.32, 2019.
- [111] A. Zhou, E. Jang, D. Kappler, A. Herzog, M. Khansari, P. Wohlhart, Y. Bai, M. Kalakrishnan, S. Levine, and C. Finn, “Watch, try, learn: Meta-learning from demonstrations and rewards”, International Conference on Learning Representations, 2020.
- [112] M. Wulfmeier, I. Posner, and P. Abbeel, “Mutual alignment transfer learning”, *Proceedings of the Annual Conference on Robot Learning*, vol.78, pp.281–290, 2017.
- [113] Y. Teh, V. Bapst, W. M. Czarnecki, J. Quan, J. Kirkpatrick, R. Hadsell, N. Heess, and R. Pascanu, “Distral: Robust multitask reinforcement learning”, *Advances in Neural Information Processing Systems*, pp.4496–4506, 2017.
- [114] B. Kartal, P. Hernandez-Leal, and M. E. Taylor, Terminal Prediction as an Auxiliary Task for Deep Reinforcement Learning, *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2019.
- [115] P. W. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, D. Kumaran, and R. Hadsell, Learning to Navigate in Complex Environments, International Conference on Learning Representations, 2017.

- [116] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, Agent Modeling as Auxiliary Task for Deep Reinforcement Learning, AAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2019.
- [117] D. Hein, A. Hentschel, T. Runkler, and S. Udluft, “Particle swarm optimization for generating interpretable fuzzy reinforcement learning policies”, Engineering Applications of Artificial Intelligence, vol.65, pp.87-98, 2017.
- [118] J. Huang, P. P. Angelov, and C. Yin, “Interpretable policies for reinforcement learning by empirical fuzzy sets”, Engineering Applications of Artificial Intelligence, vol.91, p.103559, 2020.
- [119] A. Likmeta, A. M. Metelli, A. Tirinzoni, R. Giol, M. Restelli, and D. Romano, “Combining reinforcement learning with rule-based controllers for transparent and general decision-making in autonomous driving”, Robotics and Autonomous Systems, vol.131, p.103568, 2020.
- [120] Z. Jiang, and S. Luo, “Neural logic reinforcement learning”, International Conference on Machine Learning, vol.97, pp.3110–3119, 2019.
- [121] D. Kimura, M. Ono, S. Chaudhury, R. Kohita, A. Wachi, D. J. Agravante, M. Tatsubori, A. Munawar, and A. Gray, “Neuro-symbolic reinforcement learning with first-order logic”, Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp.3505–3511, Association for Computational Linguistics, 2021.
- [122] V. Goel, J. Weng, and P. Poupart, “Unsupervised video object segmentation for deep reinforcement learning”, Advances in Neural Information Processing Systems, vol.31, 2018.
- [123] X. Wang, S. Yuan, H. Zhang, M. Lewis, and K. P. Sycara, “Verbal explanations for deep reinforcement learning neural networks with attention on extracted features”, IEEE International Conference on Robot and Human Interactive Communication, pp.1-7, 2019.
- [124] V. Zambaldi, D. Raposo, A. Santoro, V. Bapst, Y. Li, I. Babuschkin, K. Tuyls, D. Reichert, T. Lillicrap, E. Lockhart, M. Shanahan, V. Langston, R. Pascanu, M. Botvinick, O. Vinyals, and P. Battaglia, “Deep reinforcement learning with relational inductive biases”, International Conference on Learning Representations, 2019.
- [125] C. A. Cruz, and T. Igarashi, “Interactive explanations: Diagnosis and repair of reinforcement learning based agent behaviors”, IEEE Conference on Games, pp.01-08, 2021.
- [126] W. Bao, Q. Yu, and Y. Kong, “Drive: Deep reinforced accident anticipation with visual explanation”, International Conference on Computer Vision, pp.7599-7608, 2021.

- [127] Q. Zhang, X. Ma, Y. Yang, C. Li, J. Yang, Y. Liu, and B. Liang, “Learning to discover task-relevant features for interpretable reinforcement learning”, *IEEE Robotics and Automation Letters*, vol.6, no.4, pp.6601-6607, 2021.
- [128] Y. Dai, H. Ouyang, H. Y. Zheng, H. Long, and X. Duan, “Interpreting a deep reinforcement learning model with conceptual embedding and performance analysis”, *Applied Intelligence*, vol.53, pp.6936-6952, 2022.
- [129] T. Zahavy, N. Ben-Zrihem, and S. Mannor, “Graying the black box: Understanding dqns”, *International Conference on Machine Learning*, pp.1899–1908, 2016.
- [130] S. M. Lundberg, and S. I. Lee, “A unified approach to interpreting model predictions”, *Advances in Neural Information Processing Systems*, vol.30, 2017.
- [131] R. Iyer, Y. Li, H. Li, M. Lewis, R. Sundar, and K. Sycara, “Transparency and explanation in deep reinforcement learning neural networks”, *AAAI/ACM Conference on AI, Ethics, and Society*, p.144–150, Association for Computing Machinery, 2018.
- [132] L. Weitkamp, E. van der Pol, and Z. Akata, “Visual rationalizations in deep reinforcement learning for atari games”, *Artificial Intelligence*, pp.151–165Springer, 2019.
- [133] J. Wang, L. Gou, H. W. Shen, and H. Yang, “Dqnviz: A visual analytics approach to understand deep q-networks”, *IEEE Transactions on Visualization and Computer Graphics*, vol.25, no.1, pp.288-298, 2019.
- [134] M. Pan, W. Huang, Y. Li, X. Zhou, and J. Luo, “xgail: Explainable generative adversarial imitation learning for explainable human decision analysis”, *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, p.1334–1343, Association for Computing Machinery, 2020.
- [135] L. He, N. Aouf, and B. Song, “Explainable deep reinforcement learning for uav autonomous path planning”, *Aerospace Science and Technology*, vol.118, p.107052, 2021.
- [136] W. Shi, G. Huang, S. Song, Z. Wang, T. Lin, and C. Wu, “Self-supervised discovering of interpretable features for reinforcement learning”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.44, no.5, pp.2712-2724, 2022.
- [137] D. Beechey, T. M. S. Smith, and O. Şimşek, “Explaining reinforcement learning with shapley values”, *International Conference on Machine Learning*, vol.202, pp.2003–2014, PMLR, 2023.
- [138] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization”, *Computer Vision and Pattern Recognition*, pp.2921–2929, 2016.

- [139] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”, *International Conference on Computer Vision*, pp.618–626, 2017.
- [140] W. Samek, G. Montavon, A. Binder, S. Lapuschkin, and K. R. Müller, “Interpreting the predictions of complex ml models by layer-wise relevance propagation”, *arXiv preprint arXiv:1611.08191*, 2016.
- [141] T. Huber, D. Schiller, and E. André, “Enhancing explainability of deep reinforcement learning through selective layer-wise relevance propagation”, *KI 2019: Advances in Artificial Intelligence*, pp.188–202, 2019.
- [142] T. Jaunet, R. Vuillemot, and C. Wolf, “Drlviz: Understanding decisions and memory in deep reinforcement learning”, *Computer Graphics Forum*, vol.39, no.3, pp.49–61, 2020.
- [143] S. Xingjian, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. c. Woo, “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting”, *Advances in Neural Information Processing Systems*, pp.802–810, 2015.
- [144] C. Beattie, J. Z. Leibo, D. Teplyashin, T. Ward, M. Wainwright, H. Küttler, A. Lefrancq, S. Green, V. Valdés, A. Sadik, J. Schrittwieser, K. Anderson, S. York, M. Cant, A. Cain, A. Bolton, S. Gaffney, H. King, D. Hassabis, S. Legg, and S. Petersen, “Deepmind lab”, *arXiv preprint, arXiv:1612.03801*, 2016.
- [145] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “Openai gym”, *arXiv preprint, arXiv:1606.01540*, 2016.
- [146] R. Araki, T. Yamashita, and H. Fujiyoshi, “Arc2017 rgb-d dataset for object detection and segmentation”, *Late Breaking Results Poster on International Conference on Robotics and Automation*, 2018.
- [147] A. L. Maas, A. Y. Hannun, A. Y. Ng, et al., “Rectifier nonlinearities improve neural network acoustic models”, *International Conference on Machine Learning*, vol.30, no.1, p.3, 2013.
- [148] E. M. Kenny, M. Tucker, and J. Shah, “Towards interpretable deep reinforcement learning with human-friendly prototypes”, *International Conference on Learning Representations*, 2023.
- [149] R. Suzuki, A. Karim, T. Xia, H. Hedayati, and N. Marquardt, “Augmented reality and robotics: A survey and taxonomy for ar-enhanced human-robot interaction and robotic interfaces”, *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pp.1–33, 2022.
- [150] H. Hedayati, M. Walker, and D. Szafir, “Improving collocated robot teleoperation with augmented reality”, *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pp.78–86, 2018.

- [151] Z. Makhataeva, A. Zhakatayev, and H. A. Varol, “Safety aura visualization for variable impedance actuated robots”, IEEE/SICE International Symposium on System Integration, pp.805–810, 2019.
- [152] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase, “Development of human support robot as the research platform of a domestic mobile manipulator”, ROBOMECH journal, vol.6, no.1, pp.1–15, 2019.
- [153] A. Juliani, V. P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, et al., “Unity: A general platform for intelligent agents”, *arXiv preprint arXiv:1809.02627*, 2018.
- [154] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation”, IEEE transactions on pattern analysis and machine intelligence, vol.39, no.12, pp.2481–2495, 2017.
- [155] “Microsoft hololens2 docs”, <https://www.microsoft.com/it-it/hololens>.
- [156] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, “Large language models are zero-shot reasoners”, Advances in Neural Information Processing Systems, vol.35, pp.22199–22213, 2022.

研究業績一覧

学術論文

- [1] Hidenori Itaya, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi, Komei Sugiura, “Mask-Attention A3C: Visual Explanation of Action–State Value in Deep Reinforcement Learning,” IEEE Access, vol. 12, pp. 86553-86571, 2024.
- [2] Hidenori Itaya, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi, “Adaptive Selection of Auxiliary Tasks Using Deep Reinforcement Learning for Video Game Strategy,” IEEEJ Transaction on Image Electronics and Visual Computing, vol. 12, No. 1, pp. 23-31, 2024.

国際会議発表論文

- [1] Chenyu Zhang, Hidenori Itaya, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi, “High-Precision for Multi-Task Learning from In-Vehicle Camera Using BiFPN,” 35th IEEE Intelligent Vehicles Symposium (IV, Poster), 2024.
- [2] Hidenori Itaya, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi, “Auxiliary selection: optimal selection of auxiliary tasks using deep reinforcement learning,” 8th IEEEJ International Conference on Image Electronics and Visual Computing (IEVC, Oral), 2024.
- [3] Hidenori Itaya, Tom Sagawa, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi, “Visual Explanation for Cooperative Behavior in Multi-Agent Reinforcement Learning,” IEEE International Joint Conference on Neural Networks (IJCNN, Oral), 2023.
- [4] Tsuyoshi Goto, Hidenori Itaya, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi, “Solving the Deadlock Problem with Deep Reinforcement Learning Using Information from Multiple Vehicles,” 33rd IEEE Intelligent Vehicles Symposium (IV), 2022.
- [5] Hidenori Itaya, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi, Komei Sugiura, “Visual Explanation using Attention Mechanism in Actor-Critic-based Deep Reinforcement Learning,” IEEE International Joint Conference on Neural Networks (IJCNN, Oral), 2021.

- [6] Hidenori Itaya, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi, “Adaptive Selection of Auxilliary Tasks in UNREAL,” The 28th International Joint Conference on Artificial Intelligence (IJCAI) 2nd scaling-up reinforcement learning (SURL) workshop, 2019.

国内会議発表論文

- [1] 鈴木陽太郎, 板谷英典, 平川 翼, 山下 隆義, 藤吉 弘亘, “歩行者が交差点に存在するシーンにおける物体検出モデルの精度評価,” 自動車技術会 秋季大会, 2024.
- [2] 鈴木佳三, 板谷英典, 村瀬卓也, 佐々木一磨, 平川翼, 山下隆義, 藤吉弘亘, “MaskDP による事前学習のマルチドメイン拡張,” 日本ロボット学会学術講演会 (RSJ), 2024.
- [3] 板谷英典, 平川翼, 山下隆義, 藤吉弘亘, 杉浦孔明, “Action Q-Transformer による説明可能な強化学習,” 第 27 回 画像の認識・理解シンポジウム (MIRU, Poster), 2024
- [4] 大鹿海都, 板谷英典, 平川翼, 山下隆義, 藤吉弘亘, “深層強化学習における状態遷移を考慮した内発的動機付けによる探索の効率化,” パターン認識・メディア理解研究会 (PRMU), 2024.
- [5] 尹文韜, 板谷英典, 真野航輔, 平川翼, 山下隆義, 藤吉弘亘, “Transformer モデルによる自律移動の視覚的説明と拡張現実による提示,” 日本ロボット学会学術講演会 (RSJ), 2023.
- [6] 山下隆義, 板谷英典, 平川翼, 藤吉弘亘, 長瀬功児, 小山翔太郎, 井上秀雄, “車載カメラを用いた物体検出における実環境と CG 環境の一致性検証,” 自動車技術会 春季大会, 2023.
- [7] 本多航也, 板谷英典, 平川翼, 山下隆義, 藤吉弘亘, “Mask-attention 機構を導入した PPO による物体把持動作の視覚的説明,” 日本ロボット学会学術講演会 (RSJ), 2022.
- [8] 板谷英典, 平川翼, 山下隆義, 藤吉弘亘, 杉浦孔明, “A3C における Attention 機構を用いた視覚的説明,” 人工知能学会全国大会 (JSAI, Oral), 2020.
- [9] 板谷英典, 平川翼, 山下隆義, 藤吉弘亘, “UNREAL における補助タスクの適応的選択,” 人工知能学会全国大会 (JSAI, Oral), 2019.

学術表彰

- [1] 2024 年 画像電子学会, IEVC2024 Best paper award.
題目: Auxiliary selection: optimal selection of auxiliary tasks using deep reinforcement learning
- [2] 2023 年 日本ロボット学会, HSR コミュニティ優秀論文賞
題目: Transformer モデルによる自律移動の視覚的説明と拡張現実による提示

- [3] 2020年 人工知能学会, 第34回人工知能学会全国大会 学生奨励賞
題目: A3Cにおける Attention 機構を用いた視覚的説明