環境内オブジェクトが持つ行動の誘導に着目した強化学習エージェント

阪口 章弘[2,a)]　金子 知適[1,b)]

概要：一般的な強化学習におけるエージェントの観測は，単なる画像や ID の羅列のような表現を持つことが多い．しかし，実世界をモデル化した多くの評価環境において環境の構成物の多くは視覚的に意味のある形状や配置が行われており，人間であればそこから環境の目的などを読み取ることが可能である．本研究は人間のこうした常識に着目し，環境内に含まれるオブジェクトを意識的に観測し活用する手法を提案する．提案手法では出現頻度の少ないオブジェクトを重要なものと推定し，その変化を追跡する能力をベースラインに組み込んだ．Minigrid を対象に複数の環境で実験を行った結果，いずれの環境においても学習能力が向上することを確認した．

Reinforcement Learning Agent Leveraging Object-Induced Actions

Sakaguchi Akihiro[2,a)]　Tomoyuki Kaneko[1,b)]

Abstract: In typical situations in reinforcement learning, agents observe an environment as an image or an array of IDs. However, many components of environments designed for reinforcement learning have visually meaningful shapes and spatial arrangements, allowing humans to infer the objectives and other characteristics of the environment. Inspired by this type of human common-sense understanding, this study proposes a method that focuses on individual objects constituting the environment state and recognize them with a more explicit manner. By focusing on rarely occurring objects and incorporating the ability to track their changes into a baseline model, the proposed method improves learning performance in several Minigrid environments.

1. はじめに

強化学習の遠大な目標として人のサポートや人による操作の代行があり，そのような環境では人の常識にそった振る舞いが求められる．学習性能評価の題材としてしばしば使われるビデオゲーム等は人工的な環境であるが，そうした環境においても人間は環境の設計者の意図を理解し，少ない試行で初めて触れるゲーム環境の主旨や最適な行動を学習する．反対に，通常の強化学習では，エージェントは入力として与えられる画像や ID の列を方策や価値関数に直接変換し，環境に人に分かりやすいデザインのオブジェクトがあったとしても，その常識的な解釈を一切活用しない．この点に着目すると，「人間にとって意味があるデザイン」を含む環境においては，環境のデザインがもつシグナルを報酬に連携させることで，強化学習を効率化し，また人が解釈や指示をしやすいエージェントの基礎になると期待される．

本研究では，前述の「常識」の一部として，環境内オブジェクトと関連が深い行動（action）に着目する．この常識を持つエージェントは，エージェントによるインタラクトが求められるオブジェクトを予想し，適切な行動を優先して試行すると期待される．そのために，

- 環境内の重要なオブジェクトを推定する補助モジュール
- 推定した重要オブジェクトの変化を追跡する補助観測

[1] 東京大学大学院 総合文化研究科
Graduate School of Game Informatics, the University of Tokyo
[2] 東京大学大学院 学際情報学府
Graduate School of Interdisciplinary Information Studies, the University of Tokyo
[a)] sakaguchi@game.c.u-tokyo.ac.jp
[b)] kaneko@acm.org

© 2026 Information Processing Society of Japan

という能力を持つエージェントを提案する．性能評価として，Minigrid [1] で計算機実験を行い，このような能力が，既存手法よりも早く安定した学習に貢献することを確認した．

2. 関連研究

人間が持つ「常識」を再現する手法として，大規模言語モデル（Large Language Model; LLM）が注目されている．強化学習においても LLM を活用する動きは広がっており，Minigrid において強化学習エージェントに LLM を組み込んだ LLM4Teach [11] や Atari において LLM に直接タスクを解決させることを試みた Atari-GPT [8] など，様々なアプローチが行われている．しかし，大量の試行錯誤を行う強化学習に単純に LLM を組み込む運用はコストが高く，学習の毎ステップに LLM を用いるような方法は，採用できる状況が限られる．

観測された画像や動画内のオブジェクトに着目する，object-centric な技術は人に近い認識をエージェントが行うことに貢献しうる．強化学習に適用した例として，ロボットアーム制御タスクを対象とした ROLL [7] や SMORL [9] がある．これらの手法では背景とオブジェクトを分離し，正確に個々のオブジェクトを認識するところに挑戦があり，オブジェクトごとに異なる行動を誘導することはまだ行われていない．人工的なビデオゲーム環境においては UI を含め細かなオブジェクトが大量に存在する環境が多い．こうした環境においては，オブジェクトの情報を取得できるよう拡張した環境を用意する [2]，人手で observation のアノテーションを行い，セグメンテーションモデルを作成する [10] などのアプローチがみられる．人の助けを借りずにエージェントがオブジェクトを正しく認識するためにはまだ課題が多く，個別のオブジェクトに関する常識を活用することは現時点では難しいが，研究の価値がある将来の課題と位置づけられる．

環境の変化に応じた内発的報酬によってエージェントの性能を高めるアプローチには C-BET [3] や DoWhaM [6] がある．これらはいずれもエージェントがもたらした環境の変化に対し内発的報酬を与えることで探索を加速し，学習の効率化を行なっている．いずれの手法においても環境に発生した変化がどれぐらい珍しいかを基準に内発的報酬を与えており，本研究で提案するオブジェクトの希少度には着目していない．

本研究で性能比較を行った DoWhaM における内発的報酬の設計を以下に整理する．

DoWhaM では，エージェントの action が環境に変化を及ばしたことを判定するため，エージェントの行動履歴 $\mathcal{H}$ を保持し，action の累積使用回数 $U^{\mathcal{H}}(a)$ と，action によって observation に変化が生じた回数 $E^{\mathcal{H}}(a)$ を記録する．内発的報酬を計算するため，時刻 t における action a_t によって生じるボーナス $B(a_t)$ を以下で定義する．

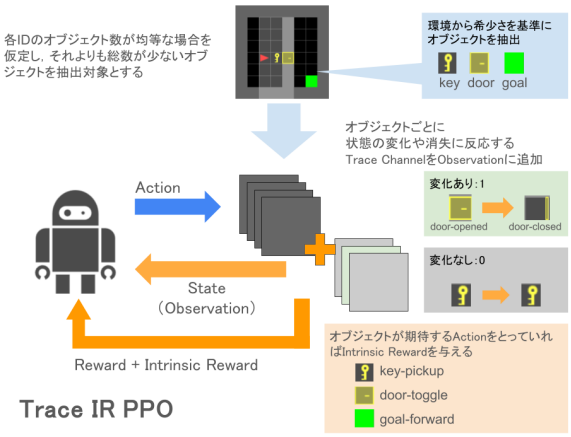


図 1: 提案手法の概要

$$B(a_t) = \frac{\eta^{1-\frac{E^{\mathcal{H}}(a_t)}{U^{\mathcal{H}}(a_t)}} - 1}{\eta - 1} \tag{1}$$

ここで η はハイパーパラメータであり，本研究では DoWhaM と同様 40 に設定している．$B(a_t)$ を用いると時刻 t における action a_t に与えられる内発的報酬は，時刻 t と $t+1$ における状態 s_t, s_{t+1} を用いて $r^i_{\mathrm{DoWhaM}}(s_t, a_t, s_{t+1})$ で定義される．

$$r^i_{\mathrm{DoWhaM}}(s_t, a_t, s_{t+1}) = \begin{cases} \dfrac{B(a_t)}{\sqrt{N^\tau(s_{t+1})}}, & s_t \neq s_{t+1}, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

ここで $N^\tau(s)$ はエピソード τ における状態 s への訪問回数である．最終的な内発的報酬は外部報酬と合算され (3) で計算される．

$$r(s_t, a_t, s_{t+1}) = r^e(s_t, a_t, s_{t+1}) + \beta r^i_{\mathrm{DoWhaM}}(s_t, a_t, s_{t+1}) \tag{3}$$

ここで β は内発的報酬のスケーリングパラメータであり，本研究では DoWhaM に準拠した 0.05 を設定している．

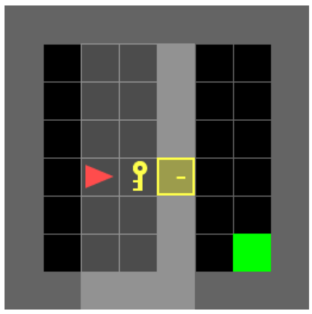
3. 提案手法

本研究では，LLM に依存しない常識として，環境内で相対的な数が少ないオブジェクト[*1]は何らかの意味を持ち，特定の行動が期待される可能性が高いオブジェクトであるという推定を取り扱う．

環境内のオブジェクトの集合を $\mathcal{O}$ であらわす[*2]．各オブジェクトの出現頻度を手がかりに，希少オブジェクトの集合 $\mathcal{O}_{\mathrm{rare}} \subseteq \mathcal{O}$ を次のように定める．

[*1] これを希少オブジェクトと呼ぶ
[*2] Minigrid の仕様上，エージェントは agent オブジェクトとして扱われるが，これはオブジェクトの種類には含まない

© 2026 Information Processing Society of Japan



wall	33
empty	27
key	1
door	1
goal	1
agent	1
total	64

図 2: key, door, goal の 3 種類が希少オブジェクトと判定

$$\mathcal{O}_{\text{rare}} = \left\{ o_i \in \mathcal{O} \;\middle|\; 0 < n_i \leq \frac{\sum_{j=1}^{sz(\mathcal{O})} n_j}{sz(\mathcal{O})} \right\} \tag{4}$$

ここで，$\mathcal{O}$ は door や wall といった環境内オブジェクトの ID の集合で，$sz(\mathcal{O})$ がその個数を表す．o_i は ID が i のオブジェクト，n_i は o_i のオブジェクト出現数を表す．オブジェクトの出現数とは，エピソード単位でエージェントが発見しうる対象全てを数える．本項が対象とする Minigrid では，同一環境であればオブジェクトの希少さはほぼ同等となるが，オブジェクトの色や種類が変化しうる環境においては議論の余地がある．また Minigrid 環境では，さまざまな色のドアや鍵が登場する．$\mathcal{O}_{\text{rare}}$ の判定では色の違いを捨象して，ドアを 1 種類，鍵をまた別の 1 種類のように数える DoorKey-8x8-v0 における希少オブジェクトのカウント例を**図 2** に示す．

次に，希少オブジェクトがエージェントの行動によって変化するかどうかを追跡する *trace channel* を，希少オブジェクト o_i ごと，色 c ごとに観測として与える．位置 (w, h) の *trace channel* の値 $\text{obs}_{\text{rare}}^{o_i,c}$ は，

$$\text{obs}_{\text{rare}}^{o_i,c}(w,h) = \begin{cases} 1, & C_{\text{dis}} \text{ or } C_{\text{chg}}, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

$\text{obs}_{\text{rare}}^{i,j}(w,h)$ はオブジェクトが初期状態から変化していなければ 0, 消失フラグ $C_{\textbf{dis}}$ や状態の変化フラグ $C_{\textbf{chg}}$[*3]があれば 1 をとる．同種類同色のオブジェクトが複数存在する場合は，そのうち 1 つでも条件を満たした時点で $\text{obs}_{\text{rare}}^{i,j}(w,h)$ を 1 にする．o_i は $\mathcal{O}_{\text{rare}}$ の要素とする，c は Minigrid のオブジェクトに設定されたカラー 6 色[*4]に対応する．対応する色のオブジェクトが実在するかどうかはエピソード毎に異なるが，汎用性のため，*trace channel* は 6 色分作成される．w, h は observation の縦横のインデックスである．

$\text{obs}_{\text{rare}}^{i,j}(w,h)$ はオリジナル環境の observation であるチャネルのスタック $\text{obs}_{\text{original}}$ の末尾に追加され，最終的に $\text{obs}_{\text{traced}}$ としてエージェントに与えられる．

$$\text{obs}_{\text{traced}} = \text{obs}_{\text{original}} \cup \text{obs}_{\text{rare}}^{o_i,c} \tag{6}$$

一般的には，オブジェクト毎に意味のある行動と意味のない行動があり，前者の方が少ないと想定される．そこで，$\text{obs}_{\text{rare}}^{i,j}(w,h)$ で追跡しているオブジェクトに対し，可能な全 action のなかからオブジェクトごとに期待する action を推定する．性能の上限を見積もるため，推定が完璧に行われた場合のオブジェクト-行動ペアを人間が用意し，オブジェクトに対して期待される action を取った時に内発的報酬を与えた．これを Trace IR と呼ぶ．内発的報酬 r_t^{int} は以下で定義される．

$$count_t = \sum_{i,j} \max_{w,h} \text{obs}_{\text{rare},t}^{o_i,c}(w,h) \tag{7}$$

$$r_t^{\text{int}} = \begin{cases} r_{\text{bonus}}, & count_t > count_{t-1} \text{ and } a_t = a'(o_i), \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

ここで $count_t$ は時刻 t において 1 になっている $\text{obs}_{\text{rare}}^{i,j}(w,h)$ のチャネル数であり，a_t は時刻 t において選択された行動 (action)，$a(o_i)$ はエージェントの正面に位置するオブジェクト o_i が期待する行動である．

Minigird におけるオブジェクトの一覧と対応する行動のペアは **表 1** のようになる．

表 1: 本研究で設定したオブジェクト–行動ペア

ID	Object	Expected Actions
0	unseen	∅
1	empty	∅
2	wall	∅
3	floor	∅
4	door	toggle
5	key	pickup
6	ball	pickup
7	box	pickup
8	goal	forward
9	lava	left, right

4. 実験と結果

今回の実験では，オブジェクトの形状的な意味が明確で，複数の object に正しい action が必要な環境として，Minigrid[1] を用いる．性能評価のベースラインとして Stable Baselines3 [4] の Proximal Policy Optimization (PPO) を使用する．

4.1 実験 1: 内発的報酬の有無による違い

まずは提案手法のうち，希少オブジェクトの変化を追跡する追加の observation と，observation の変化に応じて，

[*3] open, closed, locked の 3 状態が該当する．
https://github.com/Farama-Foundation/Minigrid/blob/b34a5755dbef71e3f1662552259a94b255c1a8bb/minigrid/core/constants.py#L42

[*4] https://github.com/Farama-Foundation/Minigrid/blob/b34a5755dbef71e3f1662552259a94b255c1a8bb/minigrid/core/constants.py#L8

© 2026 Information Processing Society of Japan

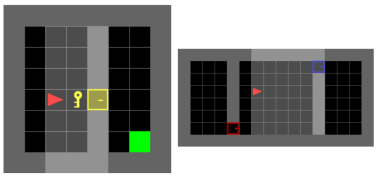
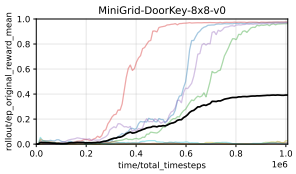
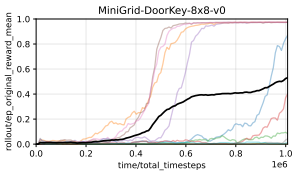


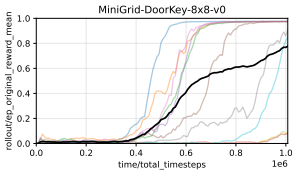
図 3: 実験 1 環境
左：DoorKey-8x8-v0 右：RedBlueDoors-8x8-v0



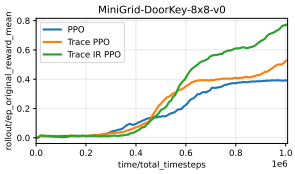
(a) PPO



(b) Trace PPO



(c) Trace IR PPO



(d) 10seed 平均の比較

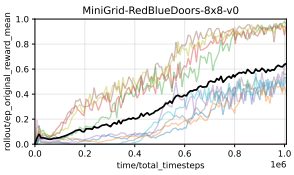
図 4: DoorKey-8x8-v0 の実験結果

正しい行動に与えられる内発的報酬の効果を検証する実験 1 を行った．この実験では，異なる行動を期待する複数のオブジェクトが存在する環境として，**DoorKey-8x8-v0**[*5]，同じ action を期待するオブジェクトが複数存在する **RedBlueDoors-8x8-v0**[*6]の 2 つを用いた．Trace IR PPO では与える内発的報酬 r_{bonus} を $1/max_time_steps$ に設定している．

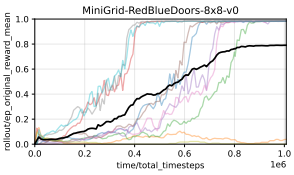
通常の PPO [5] と，検出した object ごとにその変化を追跡する *trace channel* を追加した Trace PPO，object が期待する action を実行することで内発的報酬を与える Trace IR PPO の 3 つのエージェントで実験を行った．DoorKey-8x8-v0，RedBlueDoors-8x8-v0 の双方で 100 万 step ずつ学習し，10seed における結果を比較した結果を**図 5d** と**図 5** にそれぞれ示す．グラフの黒線は 10seed の平均を示している．

両者の結果から，いずれの環境においても *trace channel* と内発的報酬を組み合わせた手法がもっとも高い成績となった．*trace channel* のみのエージェントと内発的報酬を組み合わせたエージェントを比較すると，DoorKey-8x8-v0 では明確に *trace channel* と内発的報酬の組み合わせの効果が表れたのに対し，RedBlueDoors-8x8-v0 では両者の違いは僅かなものに留まった．このことからオブジェクトや

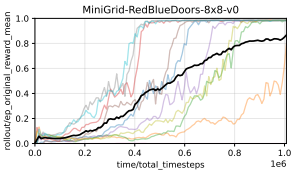
[*5] key を pickup したのち，door を toggle して通過することで goal に到達する環境
[*6] 赤い扉を先に toggle した上で青い扉を toggle することが目標の環境で，先に青い扉を toggle すると失敗扱いとなる



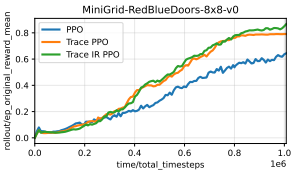
(a) PPO



(b) Trace PPO



(c) Trace IR PPO



(d) 10seed 平均の比較

図 5: RedBlueDoors-8x8-v0 の実験結果

必要な行動の種類が増加すれば，内発的報酬の効果が高まる可能性が示唆された．また，RedBlueDoors-8x8-v0 において door が期待する行動からは読み取れない，ドアを開ける順番を正常に学習できることも確認できた．

4.2 実験 2: 性能比較実験

実験 1 の結果を踏まえ，*trace channel* と内発的報酬を組み合わせた Trace IR PPO の性能を PPO, および PPO + DoWhaM[6] と比較し，Minigrid の各種環境で評価を行った．実験はいずれも 10seed 実施し，その平均の値を用いている．

本実験では，実験 1 で使用した環境に加え，
- **MiniGrid-MultiRoom-N7-S4-v0**
- **MiniGrid-MultiRoom-N12-S10-v0**
- **MiniGrid-KeyCorridorS4R3-v0**
- **MiniGrid-KeyCorridorS5R3-v0**

の 4 環境を追加して実験を行った．

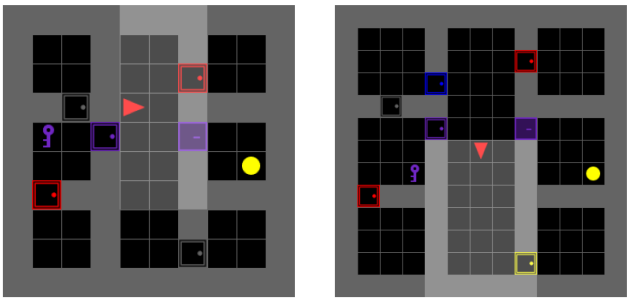
- *MiniGrid-MultiRoom*
  - 通路で繋がれた複数の部屋を通過し，ゴールに到達する環境
  - N は部屋数を示し，S は部屋の最大サイズを規定する
- *MiniGrid-KeyCorridor*
  - ロックされた扉を別の部屋にある鍵を拾って開け，扉の先のオブジェクトを拾う環境
  - S は部屋のサイズを示し，$2*R$ 個の部屋が生成される

比較対象である DoWhaM の内発的報酬は最大で 0.05 かつ，同じ状態に対して複数回の内発的報酬が発生しうるのに対し，提案手法は希少オブジェクト 1 つにつき 1 度だけ内発的報酬が発生する．これを踏まえ，本実験での提案手法の内発的報酬 r_{bonus} は 0.05 に固定している．

4.3 結果

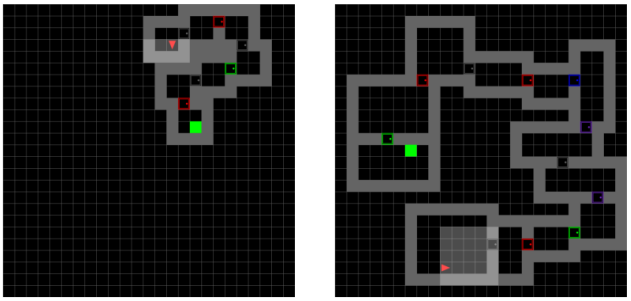
性能比較実験の結果を**図 8**–**図 13** に示す．いずれも

© 2026 Information Processing Society of Japan



(a) KeyCorridorS4R3-v0　(b) KeyCorridorS5R3-v0

図 6: MiniGrid-KeyCorridor



(a) MultiRoom-N7-S4-v0　(b) MultiRoom-N12-S10-v0

図 7: MiniGrid-MultiRoom

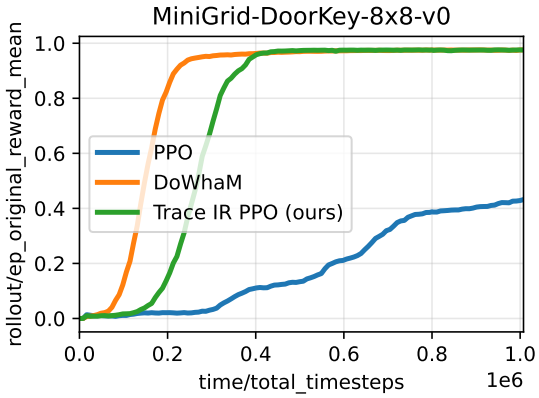


図 8: DoorKey-8x8-v0

10seed の結果を平均したものをプロットしている．ベースラインの PPO は実験 1 でも使用した DoorKey-8x8-v0 と RedBlueDoors-8x8-v0 を除くと全く学習が進まなかったのに対し，DoWhaM と Trace IR PPO は全ての環境を学習することに成功した．DoorKey-8x8-v0 と MultiRoom 環境においては DoWhaM の方が学習速度が早くなったが，今回の環境で最も複雑な KeyCorridor においては提案手法の学習スピードが大きく上回り，RedBlueDoors-8x8-v0 と KeyCorridor 環境では最終的な成績も提案手法の方がよいという結果を得た．

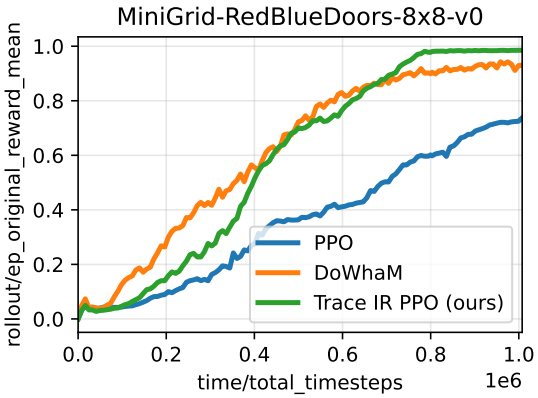


図 9: RedBlueDoors-8x8-v0

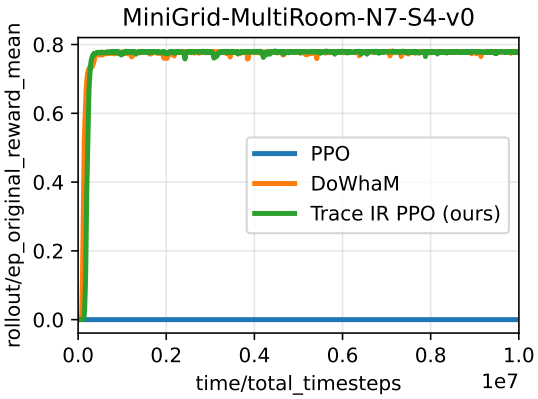


図 10: MultiRoom-N7-S4-v0

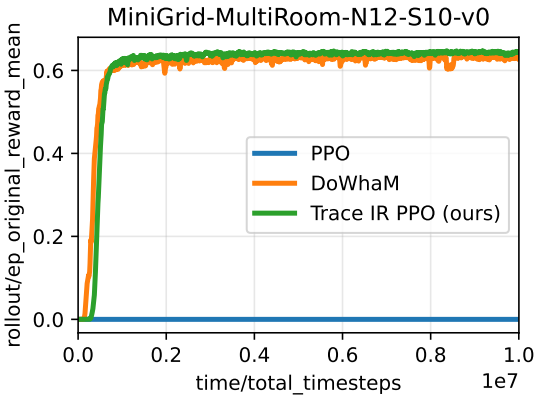


図 11: MultiRoom-N12-S10-v0

4.3.1 実験 3: 追加実験

提案手法では，基本的に内発的報酬がタスクの達成に必要な行動をとった場合にしか発生しない[*7]．対して，性能比較を行った DoWahM は何もない空間に forward するだけでも内発的報酬が発生するケースや，拾ったオブジェクトを drop し再度 pickup する行為を繰り返して高い内発的

[*7] KeyCorridor では探索の必要がない領域にもドアオブジェクトが設置されているため，それらのドアを開けた際には不必要な内発的報酬が発生しうる

© 2026 Information Processing Society of Japan

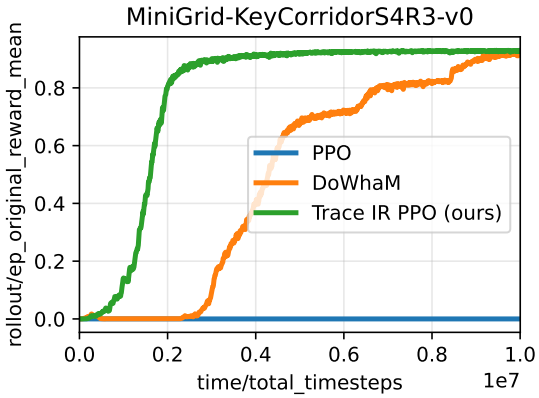


図 12: KeyCorridorS4R3-v0

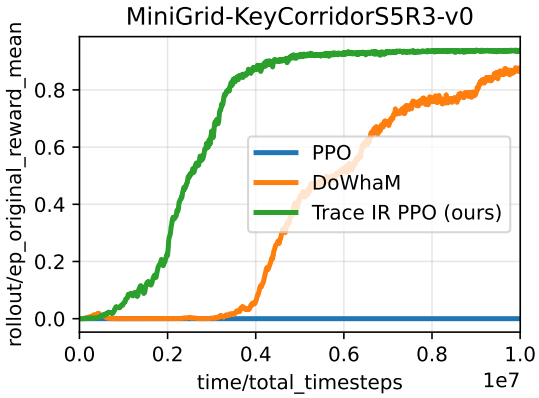


図 13: KeyCorridorS5R3-v0

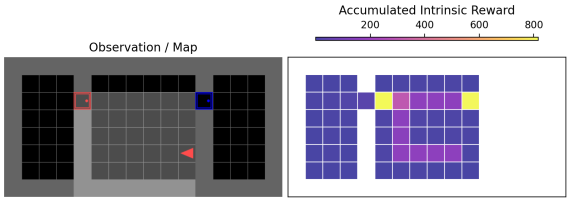


図 14: 環境を固定した RedBlueDoors-8x8-v0

報酬を得るケースが発生し，学習初期においてこれらの内発的報酬はノイズとなりうる．それでも DoorKey-8x8-v0 と MultiRoom 環境において DoWhaM の方が学習速度が早くなった理由として，これらの環境では探索の必要がない，外れの部屋のような領域がないことが挙げられる．実験 1 では DoorKey-8x8-v0 より容易な環境とみられた RedBlueDoors-8x8-v0 において DoWhaM の最終成績が提案手法を下回ったことも同様の理由と思われる．

この仮説を検証するため，DoWhaM がオブジェクトの配置を固定した環境で学習した際の累積内発的報酬をヒートマップにプロットしたものを，**図 14**–**図 16** に示す．

RedBlueDoors-8x8-v0 では，赤いドアを開けた先の領域は探索する必要がない．しかし，DoWhaM ではこうした探索にも少量の内発的報酬を発生させてしまっていることが図 14 から読み取れる．DoorKey-8x8-v0 や MultiRoom 環境においてもこうした不要な内発的報酬は発生するが，悪影響よりも隅々までの探索が動機づけされてゴールへの到達が早まる恩恵の方が大きいのに対し，RedBlueDoors-8x8-v0 や KeyCorridor 環境では無駄な探索に step を使ってしまい性能が低下したと考えられる．



図 15: 環境を固定した DoorKey-8x8-v0

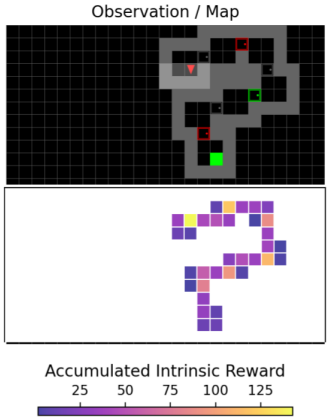


図 16: 環境を固定した MultiRoom-N7-S4-v0

こうした望ましくない内発的報酬を抑制することの効果を確かめるため，DoWhaM の内発的報酬発生条件に，エージェントの正面が希少オブジェクトであることを追加し，MiniGrid-KeyCorridorS5R3-v0 で実験した結果を**図 17** に示す．

この結果から，DoWhaM においても希少オブジェクト着目に効果があることがわかった．内発的報酬の発生頻度を抑える必要がある場合，希少オブジェクトへの着目は効果的であるといえる．

4.3.2 内発的報酬の報酬量

性能比較実験においては，DoWhaM 側の最大値に揃える形で r_{bonus} を 0.05 に固定した実験を行った．この報酬量が適切であるかどうかは検討が必要である．例えば，提案手法と DoWhaM で最も性能差があった KeyCorridorS5R3-v0 において内発的報酬を r_{bonus} を $1/max_time_steps$ に設定

© 2026 Information Processing Society of Japan

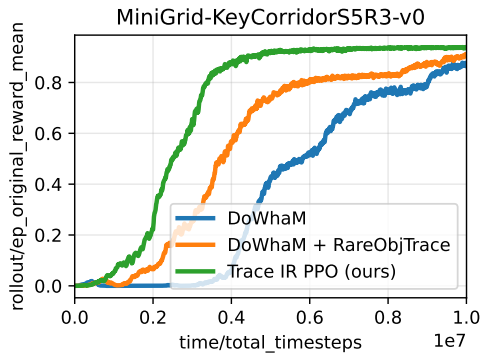


図 17: DoWhaM + 希少オブジェクト条件

すると，**図 18** のように大幅に性能が低下することが確認されている．

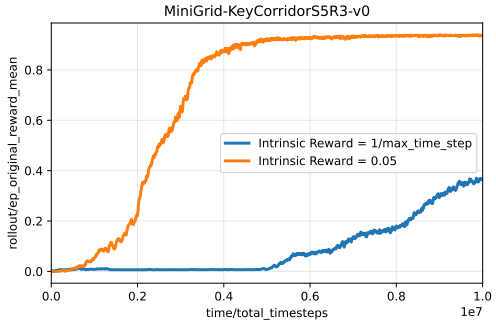


図 18: 提案手法の内発的報酬設定

5. おわりに

本稿では，人間の常識に沿った observation の認識として環境を構成するオブジェクトに着目し，その変化を追跡して内発的報酬を得るエージェントを提案した．相対的に出現数が少ないオブジェクトを対象とする手法は実験環境のいずれにおいても効果を発揮し，手法の有用性が示された．しかし，オブジェクトの認識手法や，オブジェクトの期待する action の推定，内発的報酬の量などが実験環境に依存した設計になっており，Minigrid 以外の環境に適用可能な手法に拡張することが今後の課題である．

5.1 提案手法の限界と今後の展望

本論文では，完全な observation と，存在する全オブジェクトの情報が取得可能な Minigrid で実験を行った．そのため，今回の成果は以下の点において極めて理想化されたモデルで得られたものであることに注意が必要である．

環境が静的なものである

Minigrid は一部を除き，オブジェクトがエージェントの行動と無関係に移動や追加されることがない．今回の実験で用いた環境はいずれも環境が作成された時点で全てのオブジェクトが出揃っており，1step 目の情報で希少オブジェクトを把握することが可能である．

観測の外部で全てのオブジェクト情報を取得している

今回の実験では本来の observation には含まれないオブジェクトの情報を取得している．提案手法を実用化するためには，observation 内に存在するオブジェクトを検出し，種類を特定するモジュールが必要となる．

オブジェクトと期待される行動が一対一対応している

Minigrid におけるオブジェクトは基本的に一種類の行動と対応するようになっている．例えば，door に対して実行する意味があるのは toggle のみで，pickup によって取り外したり，forward によって突き破ったりと多様なインタラクションを行うことは不可能である．そのため，オブジェクト一行動のペア推定を無視して希少オブジェクトの変化のみで内発的報酬を与えることが可能になってしまっている．ロボットが家事をするタスクのように 1 つのオブジェクトに対し，複数の行動候補が考えられるような環境での評価は，今後の課題の 1 つである．

極端にオブジェクト数が少ない環境に対応できない

提案手法で使用した希少オブジェクトの判定は，壁や床といったエージェントのインタラクションが不要なオブジェクトが重要なオブジェクトと比較して十分多いことを前提としている．これは，MiniGrid-KeyCorridorS3R3-v0 のように極端に環境が小さく床が希少オブジェクトと判定されてしまう場合や，Atari 環境における Pacman[*8]の food[*9]のように，大量の同一デザインオブジェクトが出現する環境などにおいて，希少性以外の観点が必要になるであろう．

オブジェクトが期待する action は人間による設定

本稿の実験設定では，オブジェクトが期待する action の推定については実装を行わなかった．この推定を実装するには LLM にオブジェクトと action のリストをそれぞれ渡し，対応付けをさせるなどの案が考えられる．簡易的な実験として Minigrid のオブジェクト ID と action のリストを gpt-5.6-sol に入力し，オブジェクト − action ペアを推定させた結果は**表 A·2** のようになった．オブジェクトの特徴については概ね正しい理解が出力されているものの，drop の扱い方などは実際の環境の挙動に則しておらず，内発的報酬をそのまま与えると望ましくない結果が得られる可能性が高い．

付　　録

A.1 Appendix

本文では記載しなかった実験の設定について補足する．

[*8] https://ale.farama.org/environments/pacman/
[*9] エージェントは移動によって接触し，できる限り多くの food を回収する必要がある

© 2026 Information Processing Society of Japan

実験にあたって Minigrid に予め用意された以下の Wrapper を適用した．

- OneHotPartialObsWrapper [*10]
- ImgObsWrapper [*11]

PPO に CnnPolicy を設定するため，Minigird の Custom Feature Extractor[*12]を調整し，適用している．

表 A·1: ハイパーパラメータ設定

Parameter	Default value	Used value
Policy	–	CnnPolicy
Learning rate*	3×10^{-4}	3×10^{-4}
n_{steps}	2048	512
Batch size	64	512
n_{epochs}	10	2
γ^*	0.99	0.99
λ^*_{GAE}	0.95	0.95
Clip range*	0.2	0.2
ent_coef	0.0	10^{-8}
Frame stacking	–	2
Seed	–	$0-9$

* Stable Baselines3 の初期値を使用

```
Given the following lists:
Object IDs: {0: unseen, 1: empty, 2: wall,
3: floor, 4: door, 5: key, 6: ball, 7:
box, 8: goal, 9: lava, 10: agent}.
Actions: {0: left, 1: right, 2: forward,
3: pickup, 4: drop, 5: toggle, 6: done}.
Associate each relevant object with its
expected action(s) based on its typical
function.
Return JSON.
```

図 A·1: プロンプト例

表 A·2: gpt-5.6-sol によるオブジェクト-action ペアの推定例

Object	Expected actions
empty	forward
floor	forward
door	toggle, forward
key	pickup, drop
ball	pickup, drop
box	pickup, drop, toggle
goal	forward, done
agent	left, right, forward

参考文献

[1] Chevalier-Boisvert, M., Dai, B., Towers, M., Perez-Vicente, R., Willems, L., Lahlou, S., Pal, S., Castro, P. S. and Terry, J.: Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks, *Advances in Neural Information Processing Systems 36, New Orleans, LA, USA* (2023).

[2] Delfosse, Q., Blüml, J., Gregori, B., Sztwiertnia, S. and Kersting, K.: OCAtari: Object-Centric Atari 2600 Reinforcement Learning Environments, *Reinforcement Learning Journal*, Vol. 1, pp. 400–449 (2024).

[3] Parisi, S., Dean, V., Pathak, D. and Gupta, A.: Interesting Object, Curious Agent: Learning Task-Agnostic Exploration, *Advances in Neural Information Processing Systems* (Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P. and Vaughan, J. W., eds.), Curran Associates, Inc., pp. 20516–20530.

[4] Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M. and Dormann, N.: Stable-Baselines3: Reliable Reinforcement Learning Implementations, *Journal of Machine Learning Research*, Vol. 22, No. 268, pp. 1–8 (online), available from ⟨https://jmlr.org/papers/v22/20-1364.html⟩ (2021).

[5] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O.: Proximal Policy Optimization Algorithms (2017).

[6] Seurin, M., Strub, F., Preux, P. and Pietquin, O.: Don't Do What Doesn't Matter: Intrinsic Motivation with Action Usefulness, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21* (Zhou, Z.-H., ed.), International Joint Conferences on Artificial Intelligence Organization, pp. 2950–2956 (2021).

[7] Wang, Y., Gautham, N., Lin, X., Okorn, B. and Held, D.: ROLL: Visual Self-Supervised Reinforcement Learning with Object Reasoning, *Proceedings of the 2020 Conference on Robot Learning* (Kober, J., Ramos, F. and Tomlin, C., eds.), Proceedings of Machine Learning Research, Vol. 155, PMLR, pp. 1030–1048 (online), available from ⟨https://proceedings.mlr.press/v155/wang21e.html⟩ (2021).

[8] Waytowich, N. R. et al.: Atari-GPT: Investigating the Capabilities of Multimodal Large Language Models as Low-Level Policies for Atari Games (2024).

[9] Zadaianchuk, A., Seitzer, M. and Martius, G.: Self-supervised Visual Reinforcement Learning with Object-centric Representations, *International Conference on Learning Representations*, (online), available from ⟨https://openreview.net/forum?id=xppLmXCbOw1⟩ (2021).

[10] Zhang, W., Jelley, A., McInroe, T., Storkey, A. and Wang, G.: Object-Centric World Models from Few-Shot Annotations for Sample-Efficient Reinforcement Learning, *International Conference on Learning Representations* (Vondrick, C., Hariharan, B., Raffel, C., Pinto, L., Yang, D. and Faust, A., eds.), pp. 91366–91394.

[11] Zhou, Z. et al.: Large Language Model as a Policy Teacher for Training Reinforcement Learning Agents, *IJCAI* (2024).

[*10] https://minigrid.farama.org/api/wrappers/#one-hot-partial-obs

[*11] https://minigrid.farama.org/api/wrappers/#image-observation

[*12] https://minigrid.farama.org/content/training/#create-custom-feature-extractor

© 2026 Information Processing Society of Japan