

階層的世界モデル Hieros の実証的再評価：内部表現解析と階層構造の影響分析

Empirical Re-evaluation of the Hierarchical World Model Hieros: Internal Representation Analysis and the Impact of Hierarchical Structure

三好 理輝^{*1}

Riki Miyoshi

劉 智優^{*2}

Ziwoo You

山田 達也^{*3}

Tatsuya Yamada

^{*1}ケンブリッジ大学

University of Cambridge

^{*2}電気通信大学

University of Electro-Communications

^{*3}大阪大学

The University of Osaka

階層的強化学習と世界モデルを結びつけた手法は、長期タスクの学習において期待されているが、その実用性や内部メカニズムについては十分な検証がされていない。本研究では、階層的世界モデルの代表例である Hieros に着目し、性能評価と内部状態の可視化を通じてその実態を検証した。Visual Pinpad 環境での実験により、Hieros はハイパーパラメータ設定に対して高い感度を示し、報酬設計や更新頻度の変更に対する頑健性に限界があることがわかった。また、Atari 環境での方策可視化では、高スコアを示しているにもかかわらず単純な行動パターンのみが学習されており、階層性を活かしたサブゴールの学習が実現されていないことを確認した。さらに、階層数の比較実験により、階層数の増加が学習の安定性を低下させることが確認された。これらの結果は、現在の階層的世界モデルにおいて理論的期待と実際の性能の間にギャップがあることを示しており、より頑健な階層的学習手法の必要性を示唆している。

1. はじめに

深層強化学習は多くの分野で成功を収めている一方で、大量の環境相互作用を必要とするサンプル効率の低さや、長期的依存を伴うタスクにおける学習の不安定性が依然として課題である [10, 8]。この課題に対し、環境ダイナミクスを潜在空間上に学習し、将来状態を予測するワールドモデルが提案されてきた [11, 12]。特に Dreamer 系列 [3, 4, 5] は、潜在状態空間上での想像学習により高いサンプル効率を実現している。さらに、長期的な依存関係を扱うために、時間抽象化を導入する階層的強化学習 (Hierarchical Reinforcement Learning, HRL) が研究されてきた [10, 8]。近年では、世界モデルと階層構造を統合した手法が提案されており、多時間スケールでの計画と想像を同時に実現する枠組みである Director [1] や Hieros [2] は、Dreamer などのフラットな世界モデルを上回る性能を示している点で注目されている。しかしながら、性能向上が報告されている一方で、階層的世界モデルの内部でどのような抽象表現が獲得されているのか、あるいは真に階層的な想像や長期計画が実現されているのかについての実証的検証は不十分であり、特に世界モデル自体を階層化した構造がどのように機能しているのかは明らかでない。そこで本研究では、階層的世界モデルの代表的手法である Hieros [2] に着目し、その性能評価および内部状態の可視化を通じて、階層的想像学習の実態を検証した。実験結果により、Hieros の性能がタスク特性、報酬設計、更新頻度、報酬関数における各項の係数、階層数などのハイパーパラメータに敏感であることが明らかになった。特に Atari 環境では、スコア上は高い性能を示しているものの、実際に学習された方策は単調な行動パターンに留まっていることが確認された。これらをふまえて、階層的世界モデルの現状を整理し、構造的課題を明らかにした。

2. 研究背景・目的

HRL は、エージェントが不確実な情報を集める行動をする探索において有益であると知られており [8]、効率的な報酬割り当て、転移学習、高い解釈性などといった点でも優位性があると考えられている [10]。また、世界モデルは環境もモデル化

することで高コストなエージェントと環境の相互作用を減らし、高いサンプル効率を実現する。これらを組み合わせた階層的世界モデルの研究も近年進められており、例えば Director [1] や Hieros [2] は特定の分野で既存のフラットな世界モデルを上回る性能を示している。Director は Dreamer という潜在空間での想像によって効率的な学習を行う世界モデル [3, 4, 5] を基盤として、方策を決めるモデルに Manager と Worker の二層構造を導入した手法である。Manager は長期的な潜在目標を生成し、Worker は短期的行動を出力するモデルであり、Worker の損失関数に目標達成度を組み込むことで、同時に学習を行う。Hieros は次の 4 つの点を改良するように Director を拡張した手法である：

1. 方策を決めるモデルだけでなく潜在空間モデルも階層化し、階層ごとに異なる時間スケールを扱えるようにする。
2. 階層数の三層以上への拡張も行う。
3. 潜在空間のダイナミクスの学習で再帰型状態空間モデル (Recurrent State-Space Model, RSSM) の代わりに線形状態空間モデルに基づく状態空間モデル (Simplified Structured State Space Sequence World Model, S5WM) を採用してより長期的な依存関係を学習できるようにする。
4. 時間計算量を削減するサンプリング戦略を導入する。

Hieros において、各階層ごとに以下のように定義される 3 つの報酬関数を持つ：

$$r_{\text{env}}(s_t, a_t) \quad (\text{環境報酬})$$

$$r_{\text{goal}} = \frac{g_t^T \cdot h_t}{\max(\|g_t\|, \|h_t\|)} \quad (\text{目標達成度報酬})$$

$$r_{\text{nov}} = \|h_t - g_\psi(h_t)\|_2 \quad (\text{新奇性報酬})$$

$$r_{\text{total}}(s_t, a_t, g_t) = r_{\text{env}}(s_t, a_t) + w_g \cdot r_{\text{goal}}(h_t, g_t) + w_{\text{nov}} \cdot r_{\text{nov}}(h_t)$$

ここで、 s_t は状態、 a_t は行動、 g_t はサブゴール、 h_t は観測をエンコードした潜在状態、 $g_\psi(h_t)$ は潜在空間モデルによる予測、 w_g と w_{nov} は重みづけ係数である。

連絡先: 三好理輝, rm2278@cam.ac.uk

Hieros は Atari ベンチマークにおいて DreamerV3 を上回る性能が報告されている一方で、学習された表現の解釈性やモデルサイズをスケールする方向の研究は不十分である。またその階層的な世界モデル全体において理論および評価実験は知られていない。そこで本研究では、階層的世界モデルの現状を理解し課題を特定するために、Hieros のハイパーパラメータや環境を変えた際の挙動の可視化を通じて、Hieros の性能および内部状態の評価、そしてその限界の理解を試みた。

3. 関連研究

3.1 世界モデル

世界モデルとは、環境の遷移ダイナミクスを学習し、潜在空間上で将来状態を予測することで方策学習を効率化する枠組みである [11, 12]。強化学習において環境はマルコフ決定過程 (MDP) [15] として定式化される：

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma) \quad (1)$$

ここで、 \mathcal{S} は状態空間、 \mathcal{A} は行動空間である。時刻 t における状態 s_t と行動 a_t から次時刻の状態 s_{t+1} への遷移確率を $P(s_{t+1} | s_t, a_t)$ と表す。報酬関数 $r(s_t, a_t)$ は状態行動ペア (s_t, a_t) に対して報酬を返す。 $\gamma \in [0, 1]$ は将来報酬に対する割引率を表す。世界モデルは遷移確率 P を近似することで潜在空間上で軌道生成 (imagination) を可能にする [11, 3]。この内部シミュレーションに基づく学習はサンプル効率を大きく向上させることが知られている [3]。

3.1.1 世界モデルの分類

世界モデルは潜在状態において次状態を予測することを目的とする状態予測モデル、高次元な観測空間において次の観測を予測することを目的とする観測予測モデル、そして観測を低次元の潜在空間に写像しその潜在空間上でダイナミクスを学習する状態空間モデルの 3 種類に大別される [13]。状態予測モデルと観測予測モデルは直接的な予測を行うが、学習の安定性や表現効率の面で課題があるとされており [11]、近年は特に状態空間モデルが注目されている [12]。

3.1.2 Dreamer

Dreamer 系列 [3, 4, 5] は潜在空間において環境をダイナミクスとして学習する代表的な状態空間モデルであり、RSSM を用いて観測から潜在表現を構築し、その潜在空間上で将来状態および報酬を予測する。最大の特徴は、実環境ではなく学習済み世界モデル内部で想像を行うことで将来軌道を生成し、その軌道に基づいて Actor-Critic を最適化する点にある。DreamerV2 は潜在状態を離散化している点が特徴であり [4]、DreamerV3 はスケール適応性および安定性の向上が行われている [5]。

3.1.3 TD-MPC2

Temporal Difference Learning for Model Predictive Control (TD-MPC) 2 [7] は世界モデルとモデル予測制御 (MPC) を統合した連続制御向け手法であり、TD-MPC [6] の拡張として提案された。Dreamer が潜在空間モデルにより生成された軌道に基づく方策学習を主軸とするのに対し、TD-MPC2 は学習された潜在モデルを用いてオンラインで行動系列を計画する点に特徴がある。モデル学習と時間差分学習を統合することで、連続制御タスクにおいて高い精度と安定性を示している [7]。

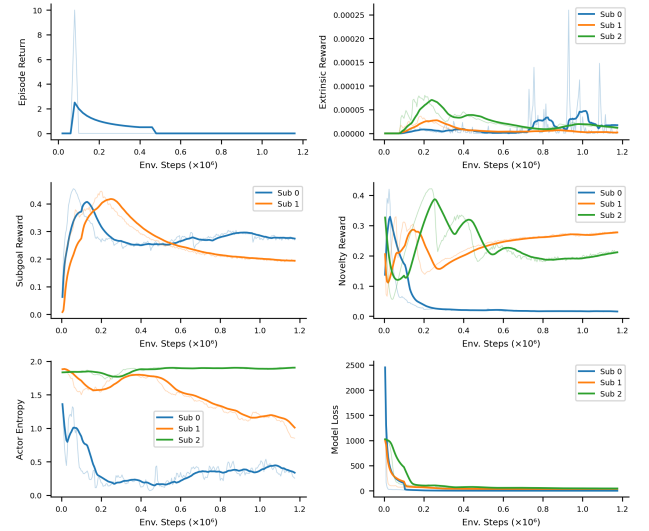


図 1: Hieros の Visual Pinpad 学習結果。ベースライン設定での学習および内部報酬曲線。

4. 実験・結果

本節では 2. で説明した Hieros [2] に関する実験について述べる。まず、検証が不十分であった、長期タスクの評価に適している Visual Pinpad 環境での検証を行った。次に、Atari でのモデルの内部状態の確認を行った。

なお、本節の実験において、Hieros のモデル検証では、既存のレポジトリ^{*1} を変更して実験を行った。本研究の実装レポジトリは^{*2} で公開している。また、Director ではこのレポジトリ^{*3} を用いて実験を行った。学習には、RTX-5070Ti または RTX-6000 単機を用いた。

4.1 Visual Pinpad

初期実験として、標準的な Pinpad 環境における Hieros のベースライン性能を評価した。Visual Pinpad は、エージェントが異なる色のタイルを特定の順番で踏むことが目的のタスクである。結果は図 1 の通りであり、400,000 ステップの学習期間を通じて累積報酬が 0 付近に留まり、有意な学習進捗が観察されなかった。これは、単純な Pinpad 設計は Hieros に対して非常に難しいタスクであることを示唆している。

4.2 Pinpad-easy

次に、モデルの変化を見やすくするため、Pinpad の報酬設計を元の実装から、末尾の一致度が上がるとスコアが与えられるように変更した Pinpad-easy というタスクを作成した。以降の実験ではこのタスクを用いる。例えば、赤緑青が正解シーケンスである Pinpad-easy-three タスクでは、赤 → 赤緑に変化した場合は +1 の報酬、赤緑 → 赤緑緑に変化した場合は -2 の報酬が与えられる。また、Hieros の内部状態について調査するために、Hieros の各レイヤが提示しているサブゴールの可視化を行った。元の Hieros の実装では過去の経験を参考にしたノイズを加えてサブゴールを可視化していたが、上位モデルの提示する情報のみに焦点を当てるために、ノイズなしで可視化した。また、Hieros エージェントの各ステップにおける、各地点への累積訪問割合をヒートマップとして可視化した。

*1 <https://github.com/Snagnar/Hieros>

*2 <https://github.com/rm-2278/Hieros-Empirical-Reevaluation>

*3 <https://github.com/danijar/director>

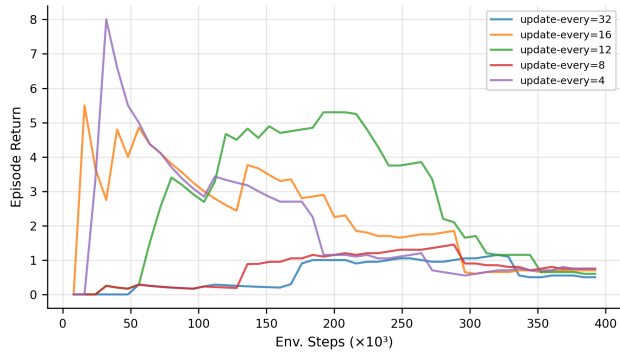


図 2: Pinpad-easy : subactor.update_every パラメータの影響。異なる更新頻度設定での学習曲線。

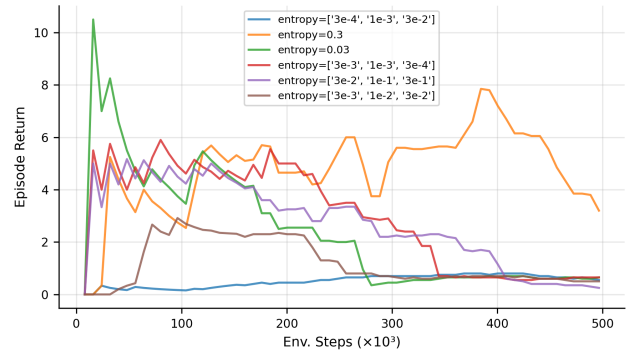


図 5: Pinpad-easy : actor_entropy パラメータの影響。異なるエントロピー設定での学習曲線。

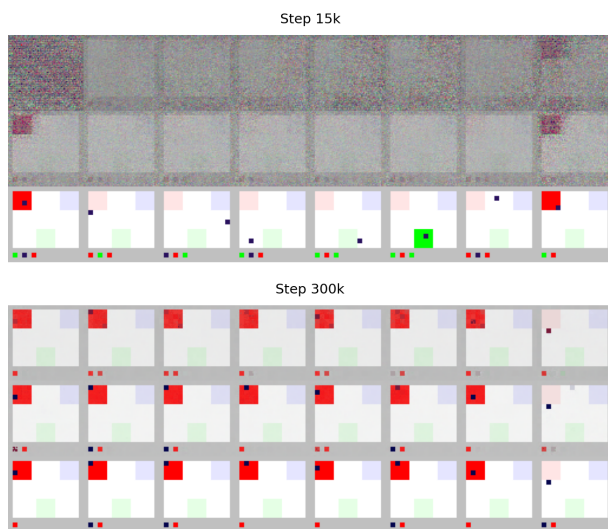


図 3: Pinpad-easy : サブゴールの時間的変化。学習過程におけるサブゴール可視化の比較 (15k ステップと 300k ステップ)。

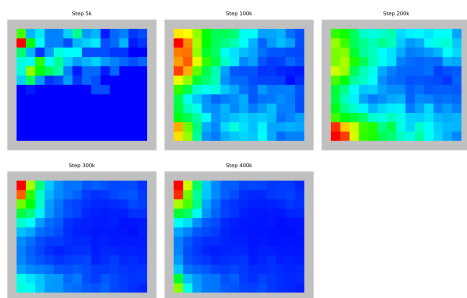


図 4: Pinpad-easy : 探索ヒートマップの時間的変化。各設定における累積訪問位置の分布。

4.2.1 サブゴール更新頻度の変更

まずは、subactor.update_every を変化させたときの結果をまとめる。subactor.update_every は、それぞれの階層がより下位の階層にどの頻度でサブゴールを提示するか、すなわち下位層をどの程度時間的抽象化しているかを表している。例えば、subactor.update_every=4 は、4 ステップごとに上位階層が下位階層に与えるサブゴールを更新していることを意味

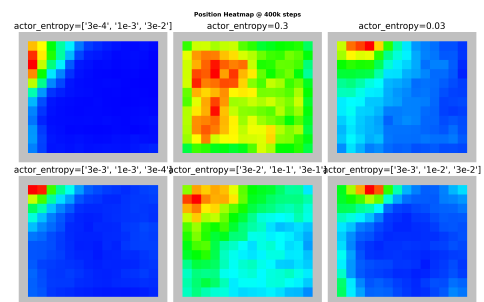


図 6: Pinpad-easy : エントロピー調整実験の探索位置ヒートマップ。各設定における探索範囲の変化。

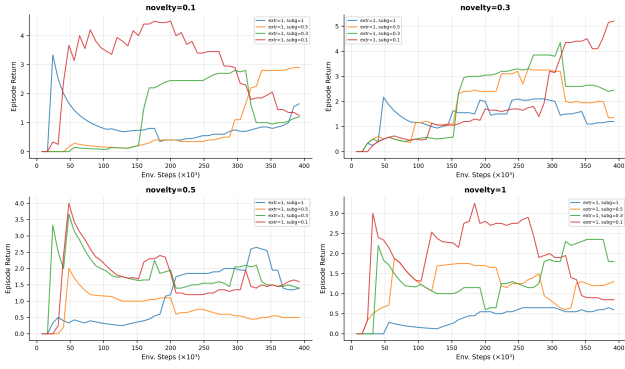
する。図 2 の定量的分析により、subactor.update_every パラメータの変動は学習過程において差異を生じるものの、最終的な収束性能は全設定において報酬値 1.0 程度の限定的な水準に留まることが確認された。また、3 において上位層が提示するサブゴールも、上位 2 層ともに赤までしか提示しておらず、求められている赤緑青のフルシーケンスを提示するには学習されなかった。同様に、4 においても探索範囲が収束している様子が確認できた。

4.2.2 方策エントロピーの変更

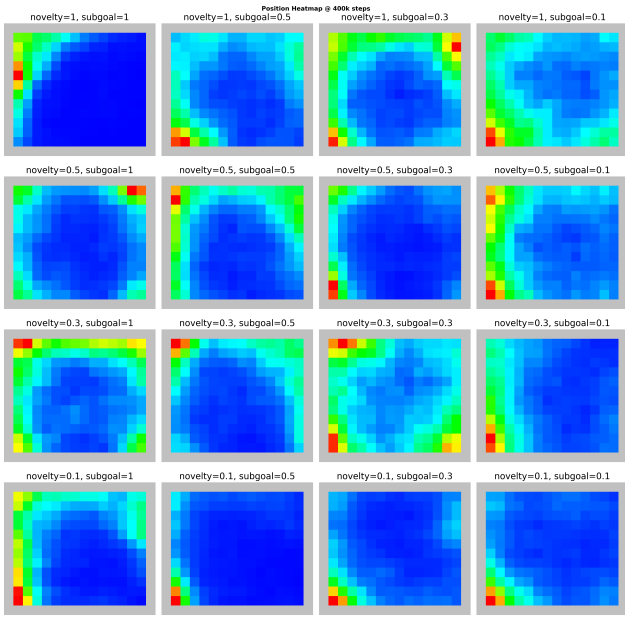
次に方策エントロピーを変更する実験の結果を図 5、図 6 に示す。図の凡例において、方策エントロピーが単一の値の場合すべての階層で同一の値であることを表し、3 つ値がある場合は左から順に下位、中位、上位階層のエントロピー値を示す。エントロピーが大きくなると、図 6 に見られるように探索範囲がより収束しにくくなり、報酬も高くなったが (図 5)、タスクを解くほどに学習が進むことはなかった。サブゴールの可視化については、図 3 と同様、赤のタイルのみ、または不適切なタイル順を示すサブゴールを提案する結果となった (図は割愛)。

4.2.3 報酬割り当て係数の変更

報酬関数における各報酬の影響を調査するために実施した実験の結果を図 7 に示す。Hieros では、各レイヤの方策は external reward, subgoal reward, intrinsic reward の 3 つの報酬を用いて学習を行っており、これらの比率を、ハイパーパラメータである各係数を変更することで調整し、学習がどう変化するかを確認した。図 7(a) に見られるように、限定的な向上が確認でき、報酬割り当て係数を変更することの有効性が示唆された。また、図 7(b) に見られるように、一部の組み合わせにおいて探索範囲が複数の角に収束することが確認され



(a) 異なる報酬割り当て係数の学習曲線。



(b) 異なる報酬割り当て係数の探索位置のヒートマップ。

図 7: 報酬割り当て係数の変更による学習への影響。400k ステップ時の探索の偏りを示している。

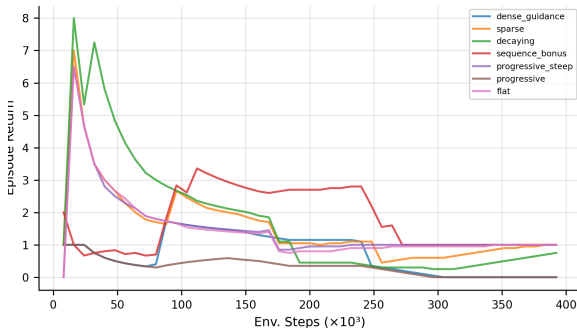


図 8: Pinpad-easy における報酬設計ごとの学習曲線。

た。一方で、他モデルからの向上は見られるが、スコアはタスクを解く水準には達しなかった。

4.2.4 Pinpad の報酬設計の変更

Pinpad-easy タスクにおいて、報酬設計を 7 種類用意し、それぞれ学習して結果を比較する実験を行った：

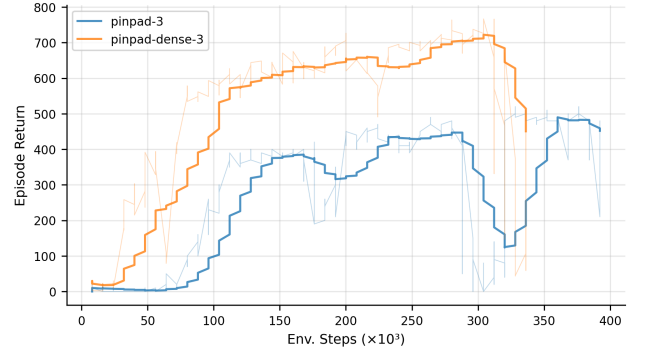


図 9: Director : Pinpad 学習結果。Pinpad および Pinpad-dense での学習曲線。

- **flat** (デフォルト): タイルを踏んだときの末尾の一致度の増減を報酬として与える。
- **progressive**: タイルを踏んだときの末尾の一致度 (suf) の増減を指数関数的にしたもの (2^{suf}) を報酬として与える。
- **sequence_bonus**: flat の報酬に加え、末尾一致度が増加ごとにボーナスで報酬を与える。
- **decaying**: 時間経過とともに報酬を減衰させる。
- **sparse**: 元の Pinpad 環境と同様に、完全なシーケンス一致時のみ報酬を与える。
- **progressive_steep**: 急勾配な指数関数的報酬 (3^{suf}) を与える。
- **dense_guidance**: flat の報酬に加えステップごとの細かい報酬 (正しい方向に進んだ場合+0.1, 間違った方向に進んだ場合-0.1) を与える。

なお、階層数は 2 にした。図 8 からわかるように、密な報酬設計にしても有意義な差異は確認されなかった。なお、図の掲載は行わないが、探索位置のヒートマップからも方策が特定領域に収束していることが確認された。

4.3 Director 実験結果

Director を用いた Pinpad および Pinpad-dense の結果を示す。Director ではどちらの報酬設計でも学習が進んでいることが確認される。Pinpad の結果は [1] で報告されている結果に近いものとなっている。また、比較のために作成した Pinpad-dense での学習も行ったところ、高い性能が得られることを確認でき、報酬設計には問題がないことが確認された。なお、GPU メモリの都合上、途中までしか学習を進められなかった。

4.4 Atari における解析

本節では、Atari ベンチマークにおける Hieros の内部挙動を解析した結果をまとめる。なお計算資源の都合上、Freeway のみは Hieros の元論文と同じ設定で、(batch size, batch length) は (16, 64)、他は (16, 16) で検証を行った。

図 10 に示すように、Hieros の Atari 環境における学習は、スコアの観点では論文どおり高い性能を示しているが、400k ステップ時点での方策画像から抽出した連続フレームを確認すると、単純な行動パターンが繰り返されていることが確認された。特に Freeway 環境においては、定量的評価指標では高性能を示しているにも関わらず、連続フレーム解析により単調な前進動作のみが確認され、複雑な方策の学習には至っていないことが明らかになった。

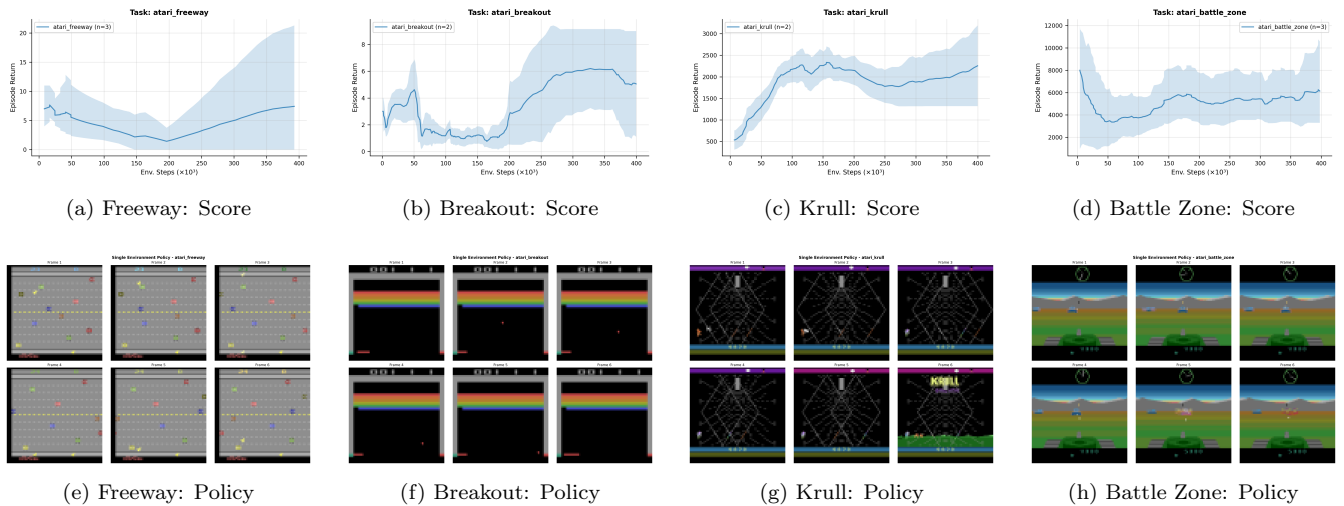


図 10: Hieros : Atari 100k ベンチマーク結果。上段：学習曲線，下段：400k ステップ時点の方策画像（5 フレーム間隔で抽出）。Freeway では単調な前進動作のみが観察される。

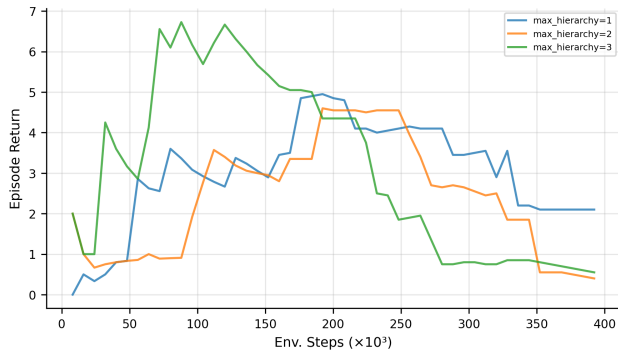


図 11: max_hierarchy パラメータの影響解析。階層数別学習曲線の比較。max_hierarchy=1 の設定でより高い性能を示している。

4.5 階層数の影響

本節では、Hieros の max_hierarchy パラメータが学習性能に与える影響を解析する。max_hierarchy パラメータは Hieros における階層数の上限を定義する重要なハイパーパラメータであり、計画の時間的抽象化レベルを決定する。図 11 に示すように、max_hierarchy=1 の設定が max_hierarchy=2,3 よりも優れた性能を示しており、階層数の増加に伴って学習性能が段階的に低下する傾向が観測された。これは付録の RSSM 結果と一致する傾向であり、階層数の増加がモデルの学習安定性を損なうことを示唆している。定量的実験結果は、理論的に予測される階層的計画手法の優位性が実際の学習動態において十分に発現されないことを示し、階層構造の導入に伴う最適化問題の複雑性増大が学習性能の制約要因となっている可能性を強く示唆している。

5. 考察・展望

本研究の包括的実験結果は、Hieros の汎化性能における根本的な限界を明らかにした。当初の仮説では、Pinpad 環境における学習困難は疎な報酬に起因すると推測していたが、密な報酬設計を導入した実験においても同様の学習停滞が観察さ

れたことから、この現象はより構造的な要因に由来すると考察できる。本稿では Atari と Visual Pinpad の 2 種類のタスクしか比較できなかったが、タスクによる学習進度の差が最も大きく、どのようなタスクで学習が進まないのか（長期的計画が求められる環境か、またはエージェントが小さい環境か）、またタスクごとに差が生まれる要因は何か（定常性の度合いか、モデルに入る外部予測誤差の大きさか）を、多様な環境下で評価することが将来の有望な方向性の一つである。

また、4.2.1、4.2.3、4.2.4 において、更新頻度や報酬割り当て係数の比率に対してモデルが敏感であることがわかった。ここから、THICK [16] のように更新頻度を可変長にすること、または HarmonyDream [9] のように報酬を自動でバランスすることで、より多様なタスクを学習できるようになると予想できる。

4.2.2 では探索範囲に変化が見られた一方、方策エントロピーを増やしてもランダム性が増えただけで、有意義な変更ではないと考えられる。

さらに Atari のモデル可視化を進める中で、有意義なサブゴールが学習されていない一方で、高スコアが出ていることが確認された。このことから、Hieros が報告する高スコアは階層性の優位性によるものではなく、環境の特性上、単純な方策でも高報酬が得られている可能性が示唆される。また、[8] で提示されているような階層化による探索の改善が関連している可能性があるが、検証が必要である。

検証結果からは Hieros がうまく学習できておらず、汎化性能も低いため、より頑健性の高いモデルの理論的・実験的模索が必要であると結論づける。

最後に、理論をコード化した際にエラーが含まれてしまった可能性は否めない一方で、世界モデルを階層化することで学習がより不安定になることが確認でき、階層性には実装によらない特有の不安定性を引き起こす要因があると考えられる。4.2.5 および Appendix にある RSSM での複数階層の実験は、これを強く示唆している。世界モデルは目的関数や誤差蓄積に起因し、また階層的モデルは階層の同時学習による非定常性によりハイパーパラメータに敏感であることが知られており、それらの掛け合わせによって学習がより一層不安定になっている可能性がある。特に、Hieros は Director と異なり潜在空間モデル

が高階層でも学習されるため、上位層の潜在空間モデルの大きな誤差が下位層の方策学習を妨げる可能性や、下位層の方策を変化させた際に上位層の潜在空間モデルのアップデートが遅れてもつれが生じる可能性が考えられる。これらの理論的・実証的解析は今後の課題である。

6. まとめ

本研究では、階層的世界モデル Hieros の性能評価と内部状態可視化を通じてその実態を検証した。主な成果は：(1) Visual Pinpad 環境では Hieros はハイパーパラメータに強く依存し、限定的な頑健性しか示さないことが明らかになった。(2) Atari 環境では高スコアにも関わらず単純な行動パターンのみが学習されており、階層性が活用されていないことが確認された。(3) 階層数の増加が学習安定性を低下し、理論上の期待と実際の性能にギャップがあることが明らかになった。

これらは、現在の階層的世界モデルに根本的な課題があることを示唆し、(1) ハイパーパラメータの頑健性向上、(2) 動的な階層的抽象化、(3) 階層化に伴う最適化の課題の理論の発展が重要であると提案する。

参考文献

- [1] Hafner, D. et al.: Deep Hierarchical Planning from Pixels, NeurIPS (2022).
- [2] Mattes, P. et al.: Hieros - Hierarchical Imagination on Structured State Space Sequence World Models (2023).
- [3] Hafner, D. et al.: Dream to Control: Learning Behaviors by Latent Imagination, ICLR (2020).
- [4] Hafner, D. et al.: Mastering Atari with Discrete World Models, ICLR (2021).
- [5] Hafner, D. et al.: Mastering Diverse Domains through World Models, Nature (2025).
- [6] Hansen, N. et al.: Temporal Difference Learning for Model Predictive Control, ICML (2022).
- [7] Hansen, N. et al.: TD-MPC2: Scalable, Robust World Models for Continuous Control, ICLR (2024).
- [8] Nachum, O. et al.: Why Does Hierarchy (Sometimes) Work So Well in Reinforcement Learning?, NeurIPS (2019).
- [9] Ma, H. et al.: HarmonyDream: Task Harmonization Inside World Models, ICML (2024).
- [10] Pateria, S. et al.: Hierarchical Reinforcement Learning: A Comprehensive Survey, ACM Computing Surveys (2021).
- [11] Ha, D. and Schmidhuber, J.: World Models, NeurIPS (2018).
- [12] Hafner, D. et al.: Learning Latent Dynamics for Planning from Pixels, ICML (2019).
- [13] Gu, A. et al.: Efficiently Modeling Long Sequences with Structured State Spaces, ICLR (2022).
- [14] Smith, J. T. et al.: Simplified State Space Layers for Sequence Modeling, ICLR (2023).
- [15] Sutton, R. S. and Barto, A. G.: Reinforcement Learning: An Introduction, MIT Press (2018).
- [16] Gumbsch, C. et al.: Learning Hierarchical World Models with Adaptive Temporal Abstractions from Discrete Latent Dynamics, EWRL (2023).

7. Appendix

7.1 RSSM

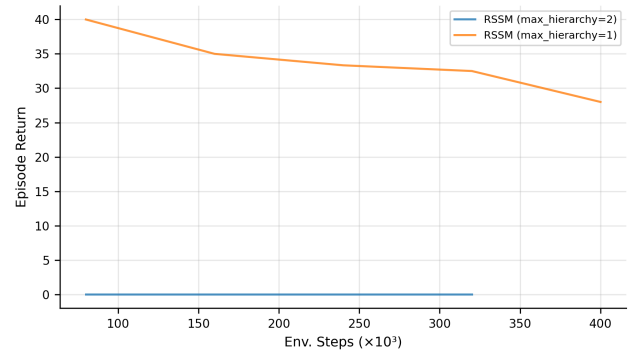


図 12: RSSM を用いた Hieros の Pinpad での学習曲線。

Hieros は S5WM を用いているが、RSSM を用いることもできる。そこで、RSSM を用いて Pinpad-easy で Hieros を学習させた結果をまとめる。図 12 で確認できるように、RSSM でも高スコアが出ることはなかった。なお、学習が進まなかったため、max_hierarchy=1 のときは 320k ステップで学習を打ち切った。

7.2 Hyperparameters

表に記載されていないものは、[2] と同じ値を使用した。

項目	値
task	pinpad-easy_three
env.pinpad-easy.reward_mode	progress_any
max_hierarchy	3
dynamics_model	s5
use_subgoal	True
novelty_reward_weight	0.1
subgoal_reward_weight	0.3
extrinsic_reward_weight	1.0
subactor_update_every	8
train_ratio	16
batch_length	32
imag_horizon	32
actor_entropy	3e-3
steps	400000

表 1: Pinpad 環境におけるハイパーパラメータの設定。