

階層的世界モデルの現状と課題：Hieros の限界と将来の展望

Current Status and Challenges in Hierarchical World Models: Identifying Limitations of Hieros and Future Prospects

三好 理輝 *¹
Riki Miyoshi

劉 智優 *²
Ziwoo You

山田 達也 *³
Yamada Tatsuya

*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、その汎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、階層的世界モデルにおける主要な先行研究の一つである 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すことがわかった。また、Atari 環境においては高スコアにも関わらず単純な方策のみが学習されており、期待する階層性を活かしたサブゴールの提案は実現されていないことを確認した。さらに、階層数の増加が学習の安定性を低下させることも明らかになった。これらの発見をもとに階層的世界モデルの今後の展望を展開する。

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 [12, 9]。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、内部モデル上で将来状態を予測する世界モデルが提案されてきた [13, 14]。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 (HRL) が研究されてきた [12, 9]。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、Director [1] や Hieros [2] は、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として報告されている。これらの手法は、ベースラインの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、階層的世界モデルが内部でどのような抽象表現を獲得しているのか、あるいは理論的に期待される階層性の利点を活用した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十分ではない。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になっていない。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、階層的世界モデルの学習の実態を検証した。これにより、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。本研究の実験結果により、Hieros の性能がタスク特性、報酬設計、更新頻度、報酬関数係数、階層数などのハイパーパラメータに依存す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を確認した。

2. 研究背景・目的

階層的強化学習 (HRL) は、探索において有益であると知られており [9]、効率的な報酬割り当て、転移学習、解釈性などといった点で優位性があると考えられている [12]。また、世界モデルは高いサンプル効率を実現し、エージェントが少ない環境との相互作用で学習することを可能にする。一方で、階層性

と世界モデルを組み合わせた研究は少ない。この2つの要素を組み合わせた数少ないモデルとして、Director [1] や Hieros [2] があり、ベースラインとして使用されている世界モデルを上回る性能を示している。

Director [1] は Dreamer を基盤とし、マネージャーとワーカーの二層方策構造を導入した手法である。マネージャーは長期的な潜在目標を生成し、ワーカーは短期的行動を出力する。具体的には、マネージャーは時刻 t における状態 s_t から潜在目標 g_t を生成し、目標達成度をワーカーの損失関数に組み込むことで、同時学習を可能にしている。Hieros [2] は Director を拡張し、

1. 方策だけでなく環境ダイナミクスを学習する内部モデルも階層化
2. 二層を超える多層構造への拡張
3. RSSM の代わりに S5WM を採用
4. 時間計算量を削減するサンプリング戦略の導入

といった改良を加えている。Hieros において、各階層毎に以下のように定義される3つの報酬関数を持つ：

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

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

$$r_{\text{nov}} = \|h_t - g_h^T(h_t)\|_2 \quad (\text{新奇性報酬})$$

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

ここで、 $f(s_t)$ は状態 s_t の特徴表現、 h_t は隠れ状態、 $g_h^T(h_t)$ はサブゴール、 w_g と w_{nov} は重みづけ係数である。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、Hieros の論文では解釈性やタスク汎用性については論じられておらず、また階層的な世界モデルの理論および評価実験も知られていない。そこで本研究では、Hieros モデル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 の性能評価および内部状態、そしてその限界の理解を試みた。

連絡先: 三好理輝, rm2278@cam.ac.uk

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、内部モデル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 [13, 14]。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：

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

ここで、 S は状態空間、 \mathcal{A} は行動空間、 $P(s_{t+1} | s_t, a_t)$ は状態遷移確率、 $r(s_t, a_t)$ は報酬関数、 $\gamma \in [0, 1)$ 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 を近似することで内部モデル上で軌道生成 (imagination) を可能にする [13, 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

世界モデルは大きく以下の 3 種類に分類できる：

- 状態予測モデル (State Transition Model)
- 観測予測モデル (Observation Prediction Model)
- 状態空間モデル (State-Space Model)

状態予測モデルは現在状態から次状態を直接予測する枠組みであり、観測予測モデルは高次元観測空間を直接予測するが、学習安定性や表現効率の面で課題がある [13]。これに対し状態空間モデルは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手法である [14]。近年の世界モデル研究の主流はこの状態空間モデルである。

3.1.2 Dreamer

Dreamer 系列 [3, 4, 5] は潜在状態空間において環境ダイナミクスを学習する代表的な状態空間モデルである。Recurrent State-Space Model (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将来軌道を生成し、imagination に基づいて Actor-Critic を最適化する点にある。DreamerV2 では離散潜在表現が導入され [4]、DreamerV3 ではスケール適応性および安定性が向上している [5]。

3.1.3 TD-MPC2

TD-MPC2 [7] は潜在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 imagination 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]。Puppeteer[8] は TD-MPC2 を階層化し、細かいマニピュレーション制御と抽象化されたアクションをそれぞれ段階的に学習している。

4. 実験

本稿では、階層的世界モデルである Hieros [2] の評価を行った。特に、論文中では検証されていなかった、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。また、Atari でのモデルの内部状態の確認を行った。

4.1 Visual Pinpad

Hieros のモデル検証では、既存のレポジトリ [10] を変更して実験を行った。また、Director ではこの既存のレポジトリ [11] を変更して実験を行った。学習には、RTX-5070Ti または RTX-6000 単機を用いた。

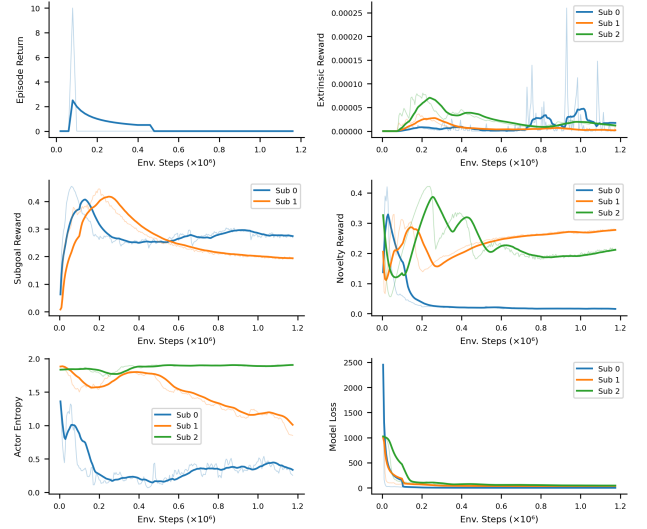


図 1: Hieros ベースコードの Pinpad 学習結果。Score および内部報酬曲線。

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。図 1 に示すように、400,000 ステップの学習を通じて累積報酬は 0 付近に留まり、有意な学習進展が観察されなかった。

4.2 Pinpad-easy

単純な Pinpad 設計では、報酬がフルシーケンスに到達するまで発生せず難しいため、モデルの変化を見やすくする目的で、以下では 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。具体例として、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。元のベースラインコード [10] 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 エージェントが各ステップにおいて、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。

4.2.1 サブゴール更新頻度の変更

次に、subactor_update_every を変化したときの結果をまとめる。subactor_update_every は、それぞれの階層が下位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 では、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という意味である。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 階層ともに赤までしか提示しておらず、求めている赤緑青のフルシーケンスを提示するには学

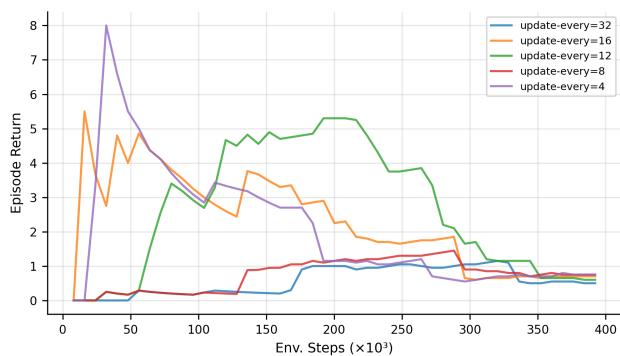


図 2: Pinpad-easy : subactor_update_every パラメータの影響。異なる更新頻度設定での学習曲線。

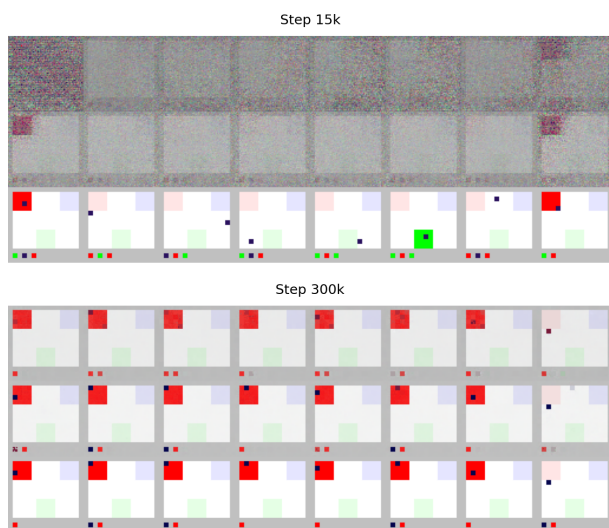


図 3: Pinpad-easy : サブゴールの時間的変化。学習過程におけるサブゴール可視化の比較 (15k ステップと 300k ステップ)。

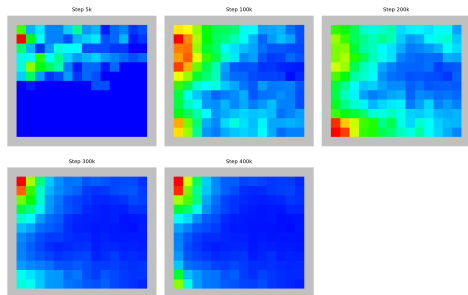


図 4: Pinpad-easy : 探索ヒートマップの時間的変化。各設定における累積訪問位置の分布。

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

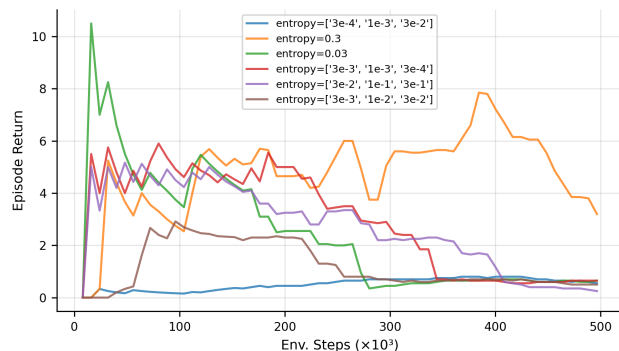


図 5: Pinpad-easy : actor_entropy パラメータの影響。異なるエントロピー設定での学習曲線。

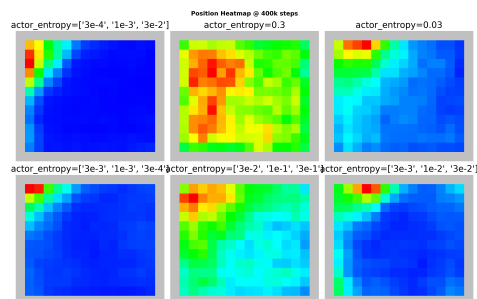


図 6: Pinpad-easy : エントロピー調整実験の探索位置ヒートマップ。各設定における探索範囲の変化。

方策エントロピーを変更したときの結果を確認した。方策エントロピーは単一の値の場合すべての階層で同一の値、3 つ値がある場合は左から順に最も下の階層、真ん中の階層、エントロピーが大きくなると、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 (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、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

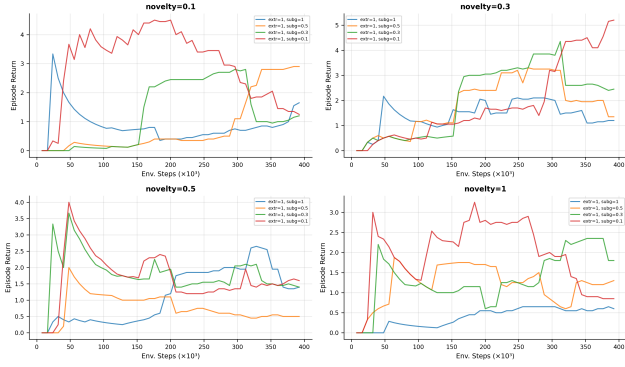
Hieros では、各レイヤの方策は external reward, subgoal reward, intrinsic reward の 3 つを用いて学習を行っている。これらの比率を、ハイパーパラメータである各係数を変更することで調整し、学習がどう変化するかを確認した。

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、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た。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。

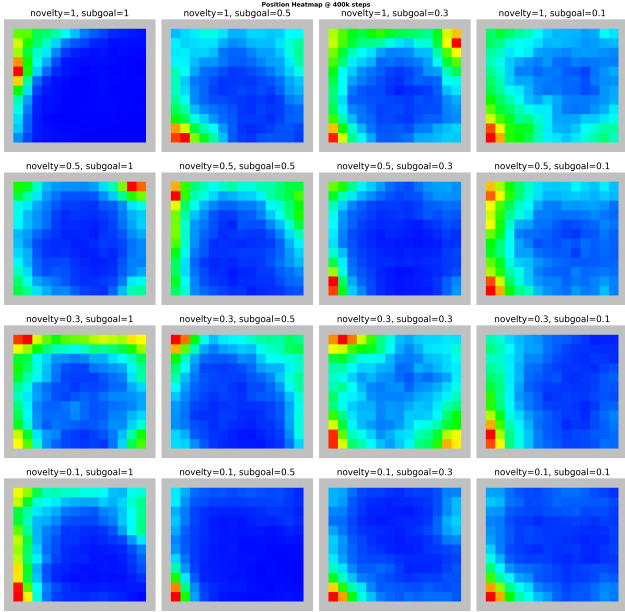
4.2.4 Pinpad の報酬設計の変更

Pinpad-easy タスクにおいて、報酬設計を変更し、学習結果を確認した。以下の 7 種類の報酬設計を比較した：

- 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が報酬
- 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が報酬 (2^{suf})



(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。

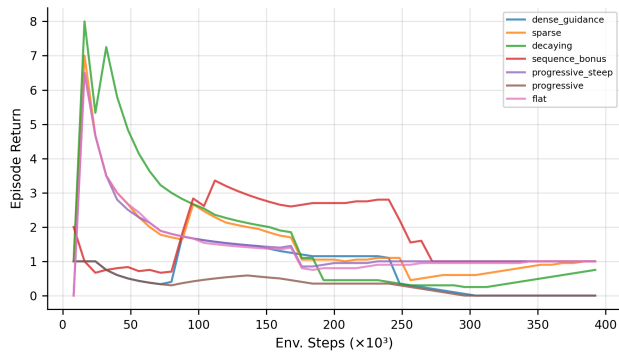


図 8: Pinpad-easy_three タスクにおける異なる報酬設計での学習曲線の比較。

- **sequence_bonus**: flat の報酬に加え、末尾一致度が増加ごとにボーナス報酬。
- **decaying**: 時間経過とともに報酬を減衰
- **sparse**: 完全なシーケンス一致時のみ報酬 (元の Pinpad)
- **progressive_steep**: 急勾配な指数関数的報酬 (3^{step})
- **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)

なお、max_hierarchy は 2 を用いた。

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、ここには掲載しないが、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

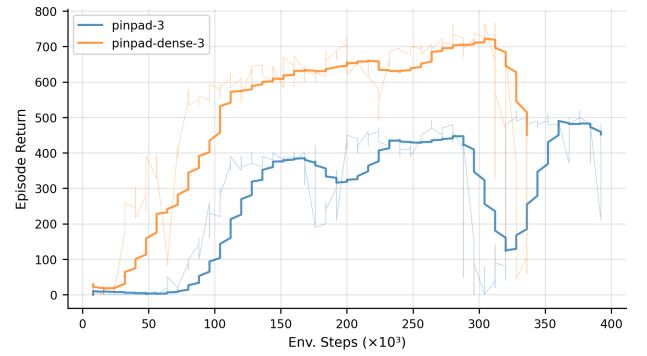


図 9: Director : Pinpad 学習結果。Pinpad-3 および Pinpad-dense-3 での学習曲線。

Director を用いた Pinpad-three および Pinpad-easy-three 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-three 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 での学習も行い、高い性能が出ることを確認した。これにより、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、途中までしか学習できなかった。

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。

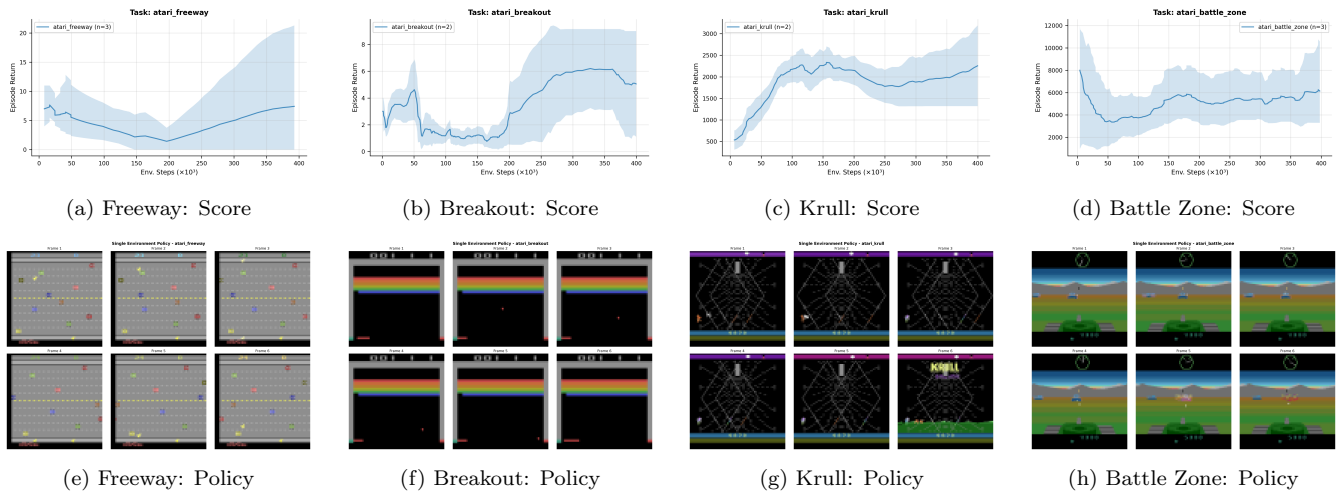


図 10: Hieros : Atari 100k ベンチマーク結果。上段：学習曲線，下段：400k ステップ時点の方策画像（5 フレーム間隔で抽出）。Freeway では単調な前進動作のみが観察される。

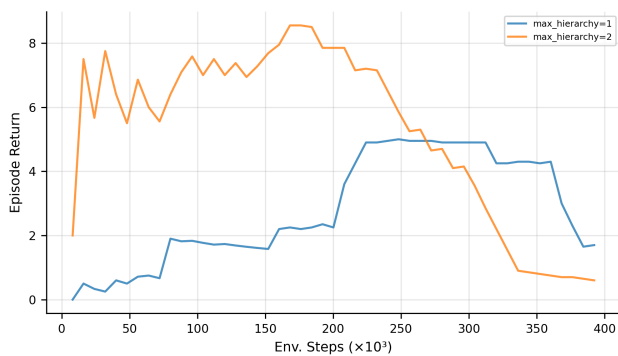


図 11: 階層数の影響

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` よりも優れた性能を示している。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結論される。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

なぜタスクごとに差が生まれるのか（定常性の大きさか，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繊細であることがわかった。ここから，VPR や THICK [17, 18] のように更新頻度を可変長にすること，また HarmonyDream [12] のように報酬を自動でバランスすることで，より多様なタスクにおいて学習できるようになると予想する。

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結論づける。

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，[9] 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もあるが，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繊細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 と異なり内部モデルが高階層でも学習されるため，この内部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，下位層が方策を変化させた際に上位内部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。(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いる。(3) 階層数の増加が学習安定性を低下させ、理論と実際の性能にギャップがある。

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, 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] Hansen, N. et al.: Hierarchical World Models as Visual Whole-Body Humanoid Controllers, ICLR (2025).
- [9] Nachum, O. et al.: Why Does Hierarchy (Sometimes) Work So Well in Reinforcement Learning?, NeurIPS (2019).
- [10] <https://github.com/Snagnar/Hieros>
- [11] <https://github.com/danijar/director>
- [12] Ma, H. et al.: HarmonyDream: Task Harmonization Inside World Models, ICML (2024).
- [13] Pateria, S. et al.: Hierarchical Reinforcement Learning: A Comprehensive Survey, ACM Computing Surveys (2021).
- [14] Ha, D. and Schmidhuber, J.: World Models, NeurIPS (2018).
- [15] Hafner, D. et al.: Learning Latent Dynamics for Planning from Pixels, ICML (2019).
- [16] Sutton, R. S. and Barto, A. G.: Reinforcement Learning: An Introduction, MIT Press (2018).
- [17] Venuto, D. et al.: Variational Predictive Routing with Nested Subjective Timescales, arXiv (2021).
- [18] Gumbsch, C. et al.: Learning Hierarchical World Models with Adaptive Temporal Abstractions from Discrete Latent Dynamics, EWRL (2023).

7. Appendix

7.1 RSSM

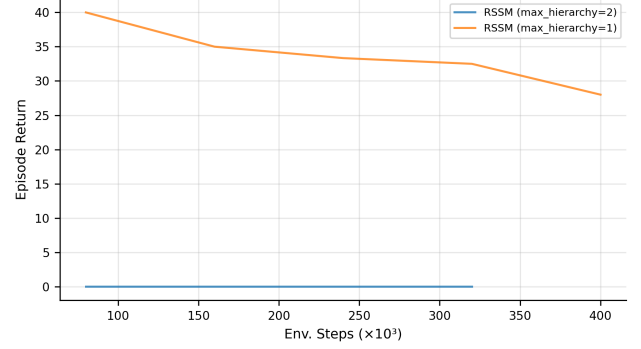


図 12: RSSM を用いた Hieros の Pinpad での学習曲線。

Hieros は S5WM を用いているが、ベースラインには RSSM も用意されている。RSSM を用いて Pinpad-easy で Hieros を学習させた結果をまとめる。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。また、S5WM を用いたとき同様階層数を少ないときにより学習が進んでいることから、階層を加えることがモデルの頑健性を下げていることがわかった。なお、学習が進まなかったため、max_hierarchy=1 のときは 320k で学習を打ち切った。

7.2 Hyperparameters

1 にあるものをベースとし、各実験において 1 つまたは 2 つの値を変更しながら組み合わせを試した。1 に記述されていないものは、元の論文 [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: Hyperparameter 設定