

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

ここに
アブストラクトを
5行程度
書き
ます

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[HRL survey, HRL benefit]。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ワールドモデルが提案されてきた[WorldModels, PlaNet]。特に Dreamer 系列 [DreamerV1, DreamerV2, DreamerV3] 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[HRL survey, HRL benefit]。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 [Director] や Hieros [Hieros] 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 [Hieros] 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。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。特に、タスクの種類および報酬設計、更新頻度、報酬関数の係数、階層数の変化に対する Hieros モデルの頑健性が低いこと、また Atari において Hieros が単調な方策およびサブゴールでも高い報酬を出せていることが確認された。

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 [HRL benefit]、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[HRL survey]。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

連絡先: 三好理輝 , rm2278@cam.ac.uk

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 [Director] や Hieros [Hieros] 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 [Director] 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。ワーカーの損失関数に目標達成度を組み込むことで、同時学習を可能にしている。

Hieros [Hieros] は Director を拡張し、

1. 方策だけでなく世界モデルも階層化
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。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。

また、解釈性やモデルをスケールさせる方向での研究はされておらず、階層的な世界モデルの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 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。

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[WorldModels, PlaNet]。強化学習において環境はマルコフ決定過程 (MDP) [SuttonBarto] 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、 $P(s_{t+1} | s_t, a_t)$ 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 [WorldModels, DreamerV1]。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でき [DreamerV1]。

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 [WorldModels]。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 [PlaNet]。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 [DreamerV1, DreamerV2, DreamerV3] 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 について 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 [DreamerV2]、DreamerV3 ではスケール適応性および安定性が向上している [DreamerV3]。

3.1.3 TD-MPC2

TD-MPC2 [TD-MPC2] 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 [TD-MPC] 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 [TD-MPC2]。

4. 実験・考察

本稿では、階層的世界モデルである Hieros [Hieros] 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 [Hieros Repo] を変更して実験を行った。また、Director ではこのレポジトリ [Director Repo] 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。

まずは、既存の Pinpad における Hieros のベースラインでの検証を行った。図 1 が示すように、報酬が 0 のまま学習が進まないことが確認された。

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（例え

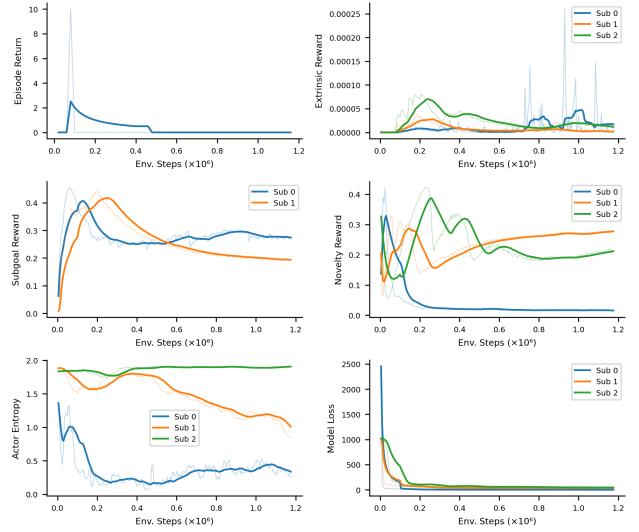


図 1: Visual Pinpad 環境における Hieros の学習および内部報酬曲線。

ば、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 [Hieros Repo] 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

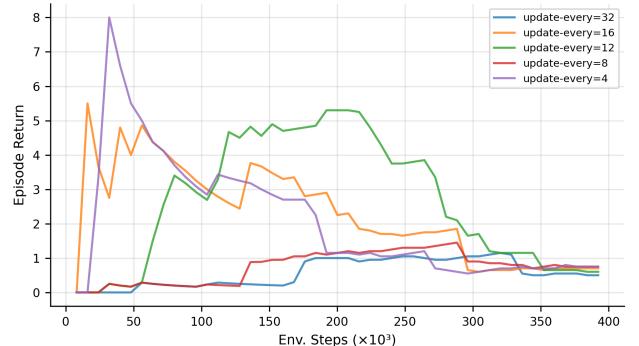


図 2: 異なる subactor_update_every パラメータでの学習曲線の比較。

次に、subactor_update_every を変化させたときの結果をまとめ。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。2 からわかるように、学習過程において多少の差異がみられる一方で、最終的には報酬が 1 程度に方策が収束す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ようには学習されなかつた。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

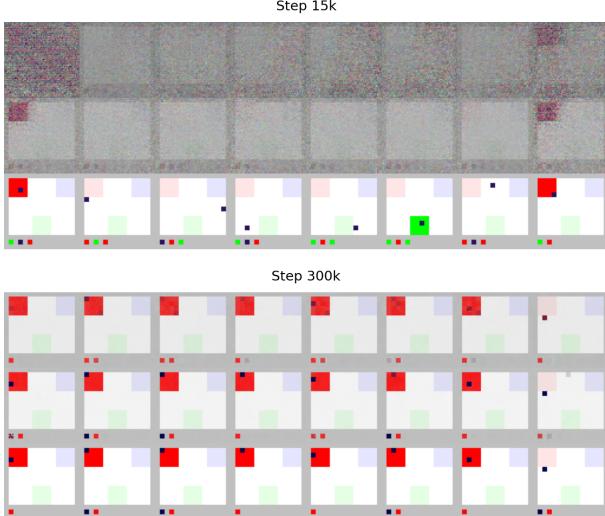


図 3: 学習過程におけるサブゴール可視化の時間的変化。典型的でわかりやすい例として 15k ステップと 300k ステップを抽出した。

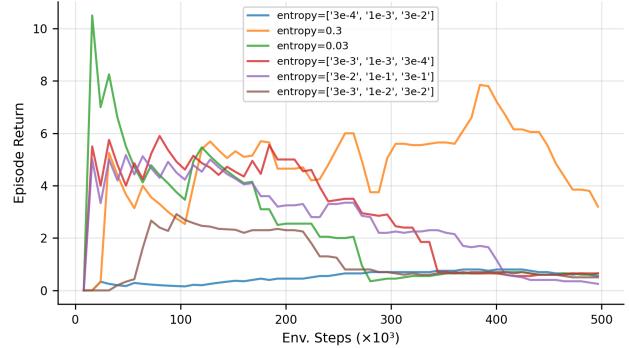


図 5: 異なる actor_entropy パラメータでの学習曲線の比較。

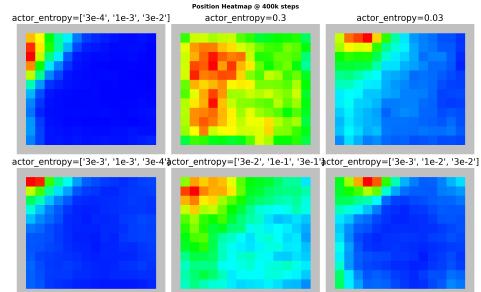


図 6: エントロピー調整実験における探索位置のヒートマップの時間的変化。

方策エントロピーを変更したときの結果を確認した。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、報酬も高くなつたが(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一一致度の増減が報酬
- progressive: タイルを踏んだときの末尾の一一致度の増減を指数関数的にしたもののが報酬 ($2^{\text{tail-match}}$)
- 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。

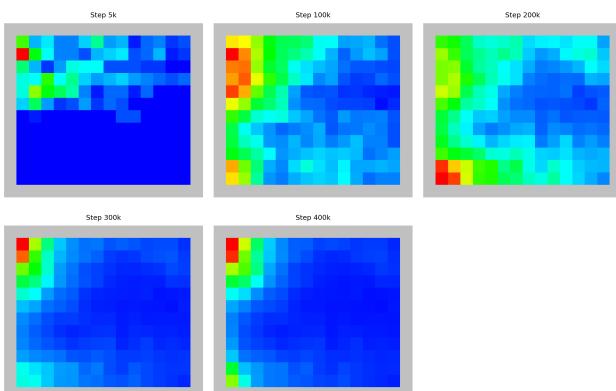
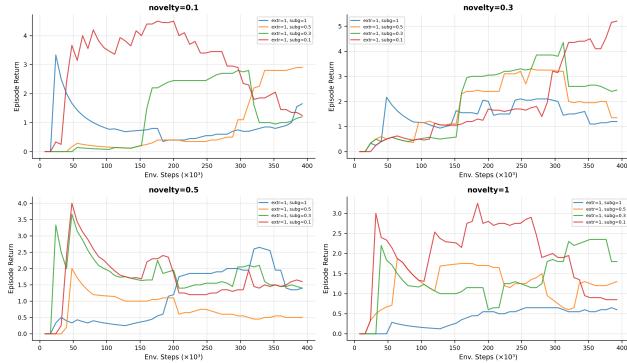
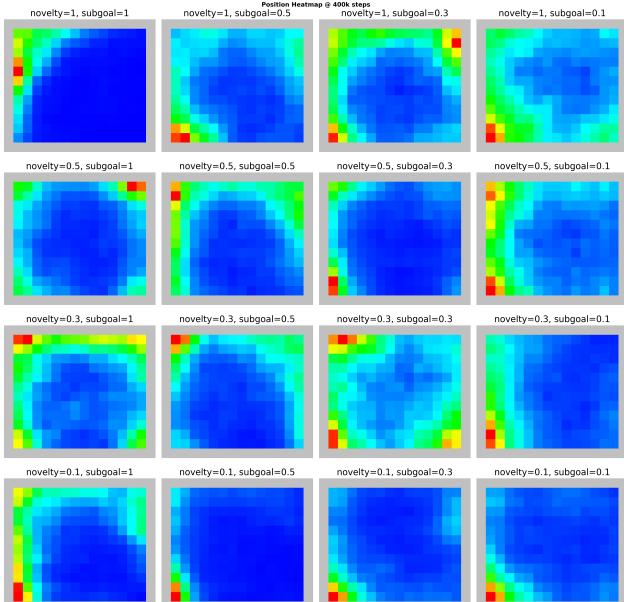


図 4: 探索位置のヒートマップの時間的変化。



(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。

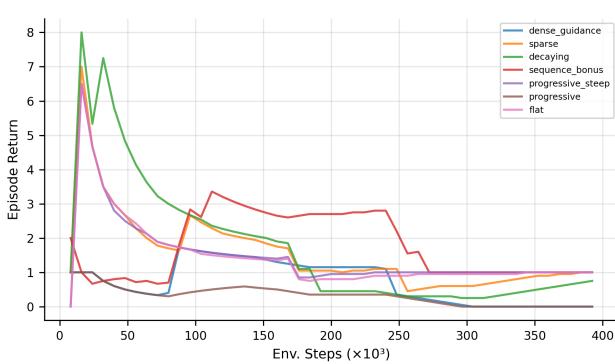


図 8: Pinpad-easy_three タスクにおける異なる報酬設計での学習曲線の比較。

- **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報酬(元の Pinpad)
- **progressive_stEEP**: 急勾配な指数関数的報酬($3^{\text{tail-match}}$)
- **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⁸、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

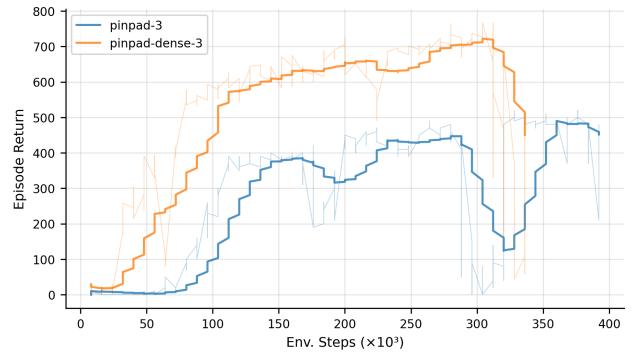


図 9: Director による Pinpad-3 および Pinpad-dense-3 の学習曲線。

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-3 の結果は [Director] 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 では、スコアは高いものの連続フレームを確認すると前進動作のみが繰り返されており、複雑な方策が学習されていない。

4.5 階層性の変更

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。

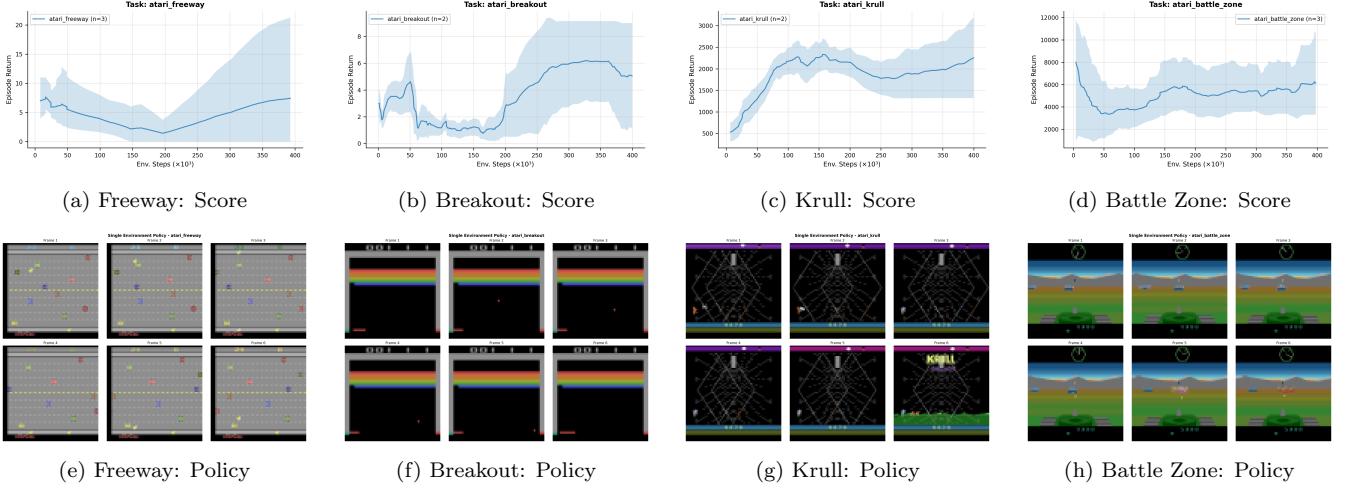


図 10: Atari 100k ベンチマークにおける学習結果。上段が学習曲線、下段が 400k ステップ時点での方策画像（5 フレーム間隔で抽出）を示す。Freeway では単調な前進動作のみが繰り返されていることが確認できる。

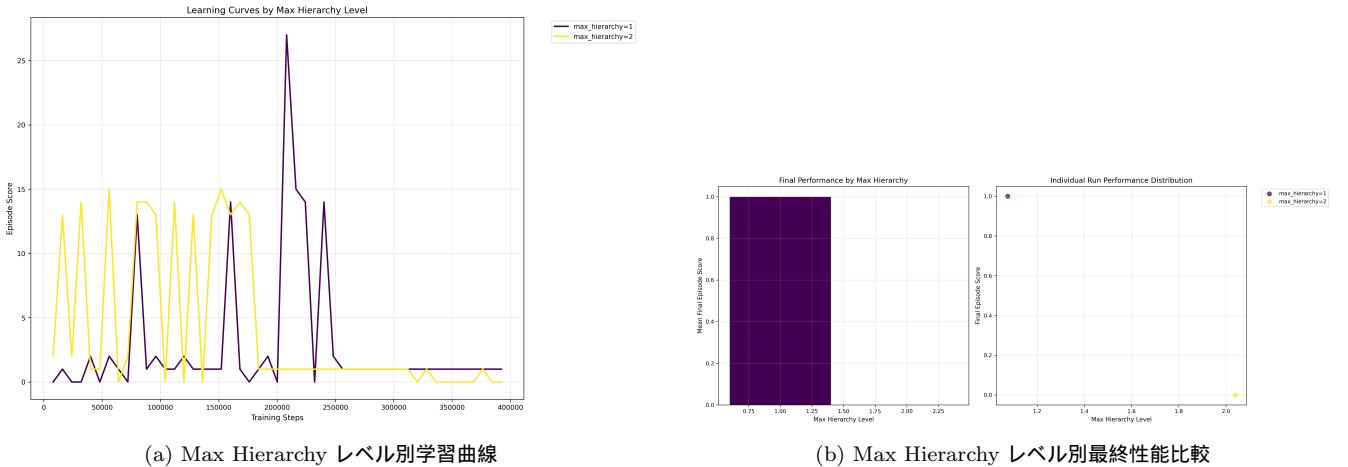


図 11: Hieros における `max_hierarchy` パラメータの影響解析。(a) 学習曲線の比較、(b) 最終性能の統計分析。`max_hierarchy=1` の設定より高い性能を示している。

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、Appendix 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。階層的計画は理論的には長期的な計画能力を向上させるはずだが、実際の学習においては最適化の困難さや過度な複雑さが性能低下を引き起こしている可能性がある。

5. 考察・展望

第 4 節全体から、Hieros のタスク間における汎化性能の低さが示された。実験当初は Pinpad で学習が進まない理由を「報酬が疎であるため」と考えていたが、Pinpad の報酬設計を密にした場合にも学習が進まなかつたため、そうではなく他の要因が大きいと考えられる。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、更新頻度を可変長にすること、また HarmonyDream [HarmonyDream] 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、[HRL benefit] 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

1
2
3
4
5
6
7
8
9
10

参考文献

- [Director] Hafner, D. et al.: Deep Hierarchical Planning from Pixels, NeurIPS (2022).
- [Hieros] Mattes, et al.: Hieros - Hierarchical Imagination on Structured State Space Sequence World Models (2023).
- [DreamerV1] Hafner, D. et al.: Dream to Control: Learning Behaviors by Latent Imagination, ICLR (2020).
- [DreamerV2] Hafner, D. et al.: Mastering Atari with Discrete World Models, ICLR (2021).
- [DreamerV3] Hafner, D. et al.: Mastering Diverse Domains through World Models, Nature (2025).
- [TD-MPC] Hansen, N. et al.: Temporal Difference Learning for Model Predictive Control, ICML (2022).
- [TD-MPC2] Hansen, N. et al.: TD-MPC2: Scalable, Robust World Models for Continuous Control, ICLR (2024).
- [Puppeteer] Hansen, N. et al.: Hierarchical World Models as Visual Whole-Body Humanoid Controllers, ICLR (2025).
- [HRL benefit] Nachum, O. et al.: Why Does Hierarchy (Sometimes) Work So Well in Reinforcement Learning?, NeurIPS (2019).

- [THICK] Gumbesch, C. et al.: Learning Hierarchical World Models with Adaptive Temporal Abstractions from Discrete Latent Dynamics, EWRL (2023).
- [Hieros Repo] <https://github.com/Snagnar/Hieros>
- [Director Repo] <https://github.com/danijar/director>
- [HarmonyDream] Ma, H. et al.: HarmonyDream: Task Harmonization Inside World Models, ICML (2024).
- [HRL survey] Pateria, S. et al.: Hierarchical Reinforcement Learning: A Comprehensive Survey, ACM Computing Surveys (2021).
- [WorldModels] Ha, D. and Schmidhuber, J.: World Models, NeurIPS (2018).
- [PlaNet] Hafner, D. et al.: Learning Latent Dynamics for Planning from Pixels, ICML (2019).
- [S4] Gu, A. et al.: Efficiently Modeling Long Sequences with Structured State Spaces, ICLR (2022).
- [S5] Smith, J. T. et al.: Simplified State Space Layers for Sequence Modeling, ICLR (2023).
- [S4WM] Deng, F. et al.: S4WM: Structured State Space World Models, arXiv (2024).
- [SuttonBarto] Sutton, R. S. and Barto, A. G.: Reinforcement Learning: An Introduction, MIT Press (2018).

7. Appendix

7.1 RSSM

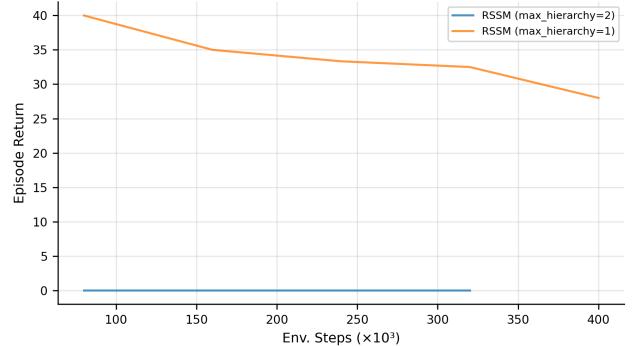


図 12: RSSM を用いた Hieros の Pinpad での学習曲線。

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時により学習が進んでいくことから、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 [Hieros] 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
decompress_subgoal_for_input	False
novelty_only_higher_level	True
novelty_reward_weight	0.1
subgoal_reward_weight	0.3
extrinsic_reward_weight	1.0
replay_temperature	0.3
subactor_update_every	8
train_ratio	16
batch_length	32
imag_horizon	32
actor_entropy	3e-3
steps	400000

表 1: Hyperparameter 設定