

階層的世界モデルにおける現状と課題：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} 劉 智優 ^{*2} 山田 達也 ^{*3}

Riki Miyoshi Jiu You 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

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。一方で、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 の性能評価および内部状態の理解を試みた。

3. 関連研究

3.1 世界モデル

世界モデルは、「説明・数式」。世界モデルは大きく、1. 状態空間モデル、2. 状態予測モデル、3. 観測予測モデル に分けることができる。代表的な世界モデルとして、Dreamer および TD-MPC2 が挙げられる。Dreamer は、状態空間モデルの一つであり、「説明」。

3.2 階層的強化学習 (HRL)

3.3 階層的世界モデル (HWM)

階層的世界モデルも、世界モデルと同様、状態空間モデルである Dreamer の方策を階層化したものが Director である。マネージャーとワーカーの 2 つの方策が存在し、マネージャーはより長い時間軸で目標とする状態を圧縮された形でワーカーに渡し、ワーカーは通常の強化学習と同様の短い時間軸で行動を出力する。ワーカーの損失関数にマネージャーの提示した目標がどの程度達成されたかを組み込むことで、ワーカーとマネー

連絡先: 三好理輝 , rm2278@cam.ac.uk

ジャーの 2 つの方策がそれぞれ役割分担しながら同時に学習できるようにしている。

Director の拡張として Hieros がある。Hieros では、

1. 方策だけでなく世界モデルも階層化している。
2. 2 階層だけでなくそれより多い階層でも実験を行っている。
3. 計算効率を上げるために、状態空間モデルとして RSSM ではなく S5WM を用いている。
4. リプレイバッファからのサンプリングを工夫し、偏りを減らすことと時間計算量の削減を実現している。

という工夫が加えられたモデルが提案されており、Atari において DreamerV3 を超える性能を示した。本稿では方策だけでなく世界モデルも階層化しているという点で Hieros に着目し、その限界と可能性を調べる。

THICK も、Director をベースとして提案された手法である。「説明」。

状態予測モデルである TD-MPC2 を階層的にしたものが Puppeteer である。これは、「説明」

4. 実験・考察

本稿では、階層的世界モデルの Hieros[Hieros] モデルの評価を行った。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 手法

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。RTX-5070Ti または RTX-6000 一つを用いて学習をした。

Pinpad における Hieros の学習状況を調査した。まずは、Hieros のベースラインでの検証を行った。

図からわかるように、報酬が 0 のまま学習が進まないことが確認された。そこで、以下では pinpad の報酬設計を元の実装から、末尾の一一致度が上がるとスコアが与えられるように変更した（例えば、赤 → 赤緑に変化した場合、+2 の報酬、赤緑 → 赤緑赤に変化した場合、-1 の報酬が与えられる）。

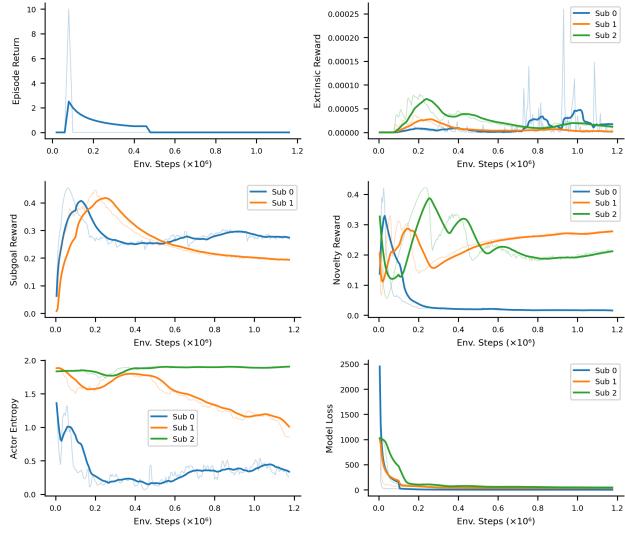


図 1: Visual Pinpad 環境における Hieros の学習および内部報酬曲線。

より詳しく Hieros の性能について調査するために、Hieros の各レイヤが提示している状態の可視化を行った。元の論文では過去の経験を参考にしたノイズを加えての可視化を行っていたが、モデルの予測のみに焦点を当てるために、ノイズなしでの可視化を行った。update-every は 8 で実験を行った。

4.2 時間的抽象化の調整

update-every は、それぞれの階層が、下の階層と比較してどの程度時間的抽象化を行っているかを表している。例えば、update-every=4 では、4 ステップごとに上の階層が下の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。

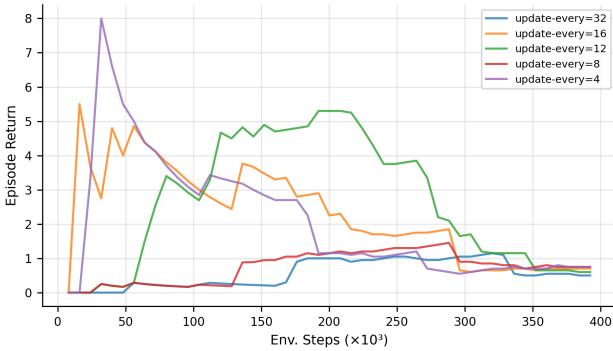


図 2: 異なる subactor-update-every パラメータでの学習曲線の比較。

図からわかるように、多少の差異がある一方で、最終的に報酬が 1 になる程度に方策が収束している。また、提示されているサブゴールも、赤までしか学習されておらず、求めている赤緑青のフルシーケンスは学習されなかった。

4.3 報酬割り当て係数の変更

Hieros では、各レイヤごとの方策は、external reward, sub-goal reward と intrinsic reward の 3 つを用いて学習を行っている。これらの比率を変更することによって、学習がどう変化するかを確認した。

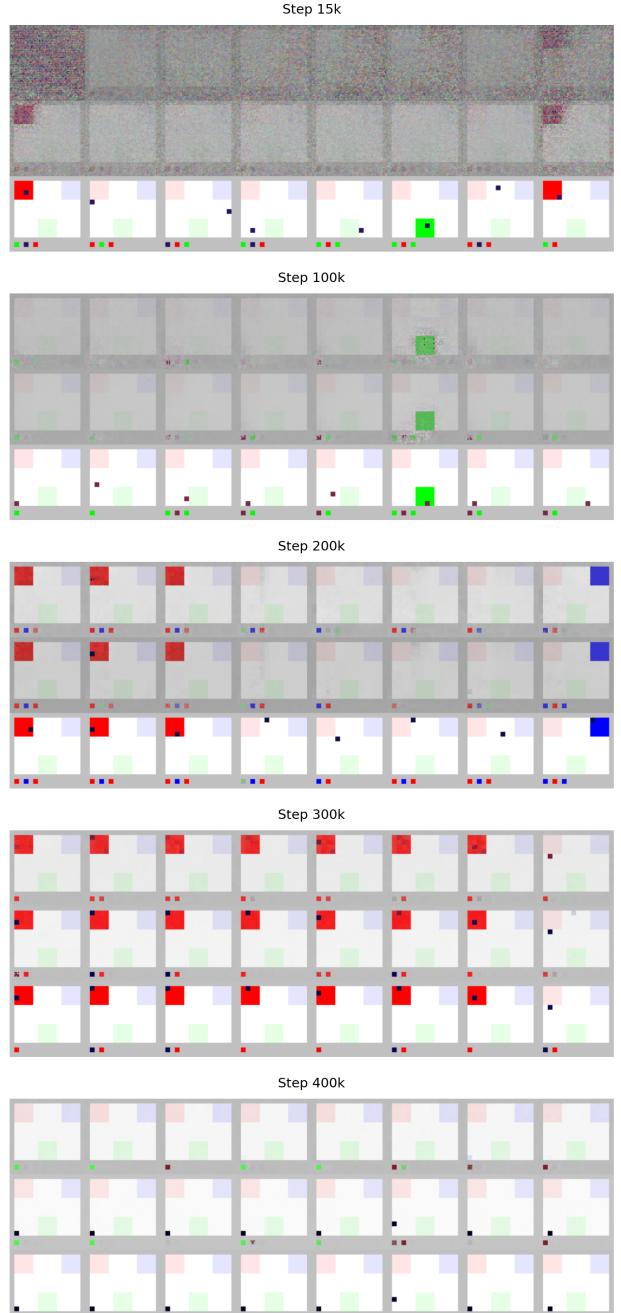


図 3: 学習過程におけるサブゴール可視化の時間的变化。

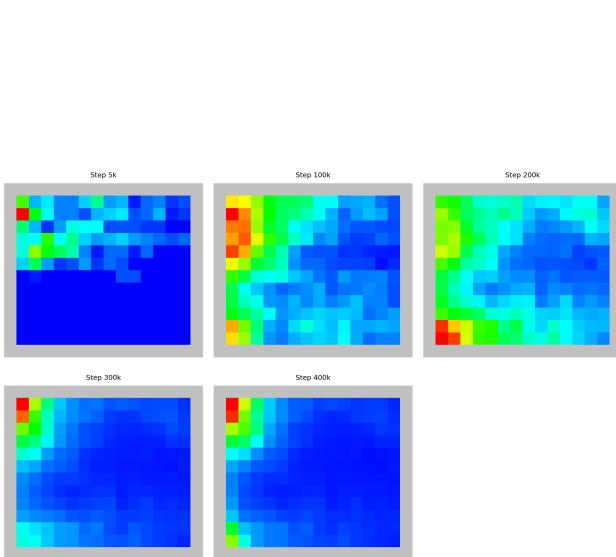


図 4: 探索位置のヒートマップの時間的变化。

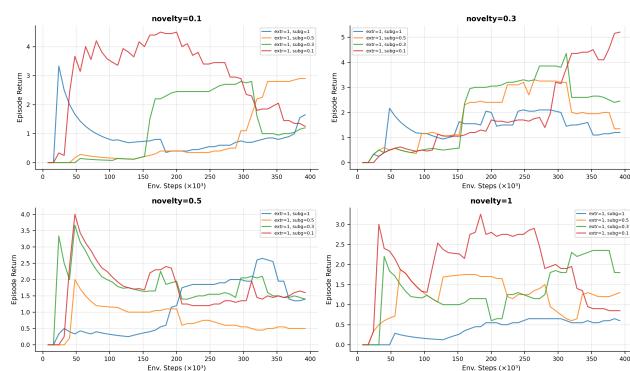
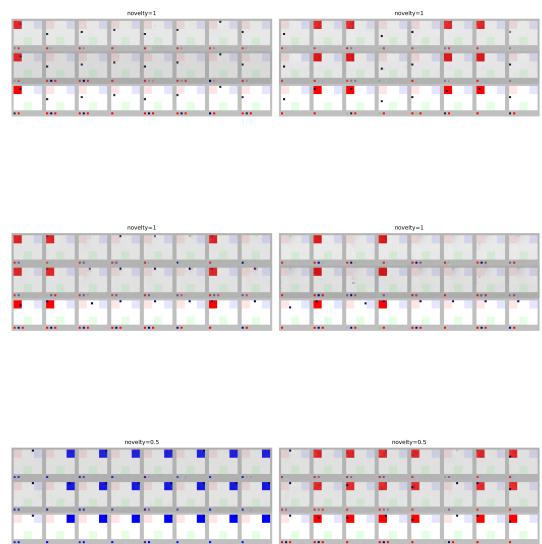


図 5: 異なる報酬割り当て係数での学習曲線の比較 (novelty_reward_weight ごとに分割)。

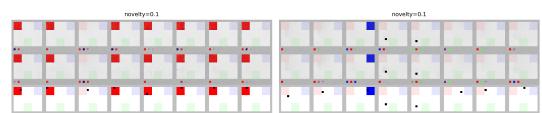
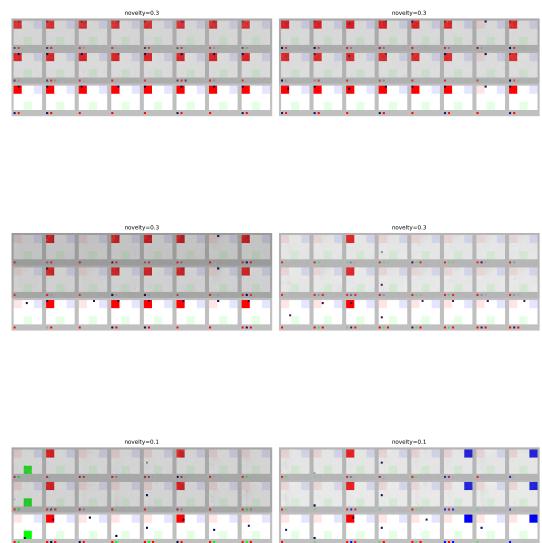


図 6: 報酬割り当て実験におけるサブゴール可視化の時間的变化。

限定期的な向上が確認でき、報酬割り当て係数を変更するとの有効性が示唆される。

4.4 方策エントロピーの変更

方策エントロピーを変更したときの結果を確認した。

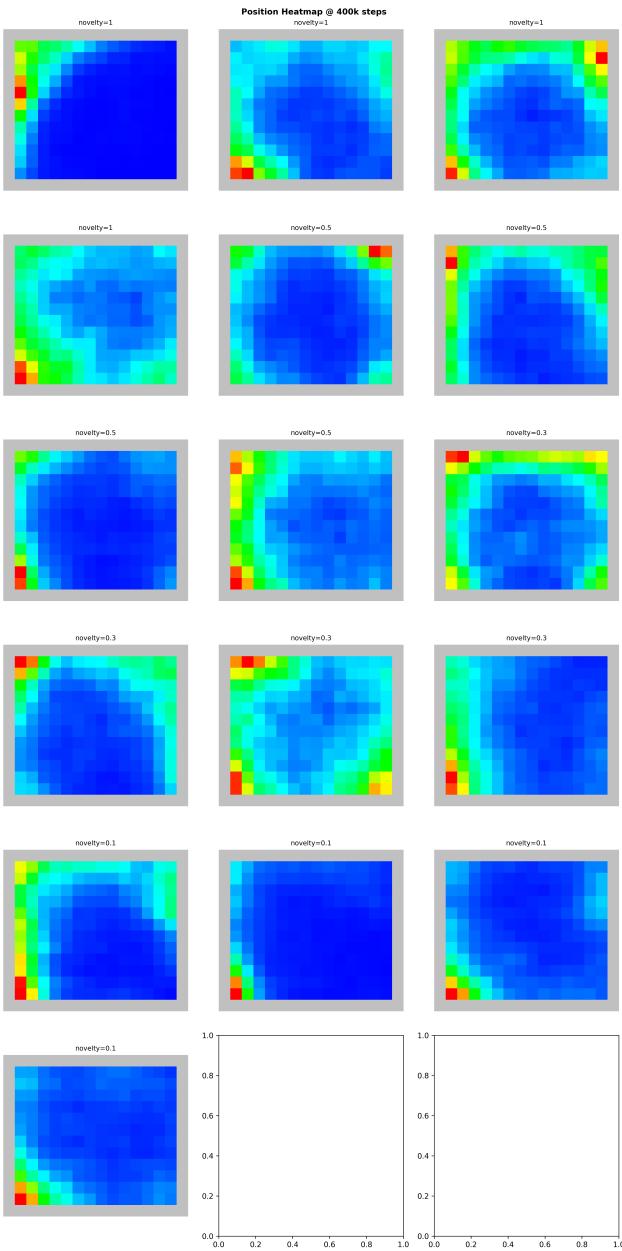


図 7: 報酬割り当て実験における探索位置のヒートマップの時間的变化。

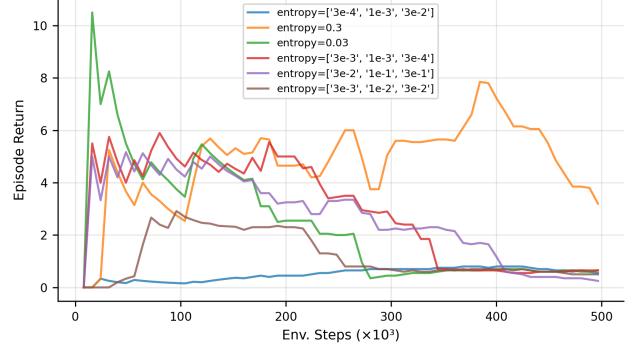


図 8: 異なる actor_entropy パラメータでの学習曲線の比較。

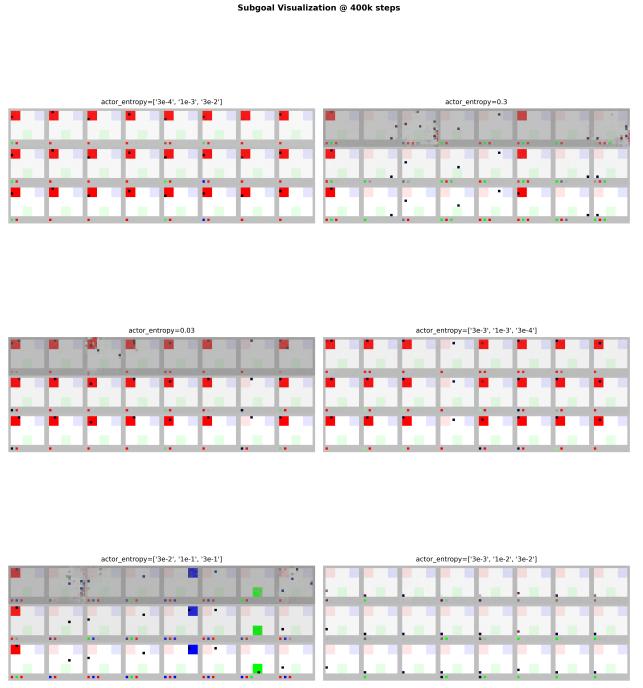


図 9: エントロピー調整実験におけるサブゴール可視化の時間的変化。

エントロピーが大きくなるとより収束しにくくなり報酬も高くなったが、顕著に学習が進むことはなかった。

4.5 Pinpad の報酬設計の変更

pinpad-easy-three タスクにおいて、報酬設計を変更した実験結果。以下の 8 種類の報酬設計を比較した：

- flat (デフォルト) : スコア増加ごとに一律+1.0 の報酬
- progressive: スコア増加ごとに指數関数的な報酬 (2^{pos})
- sequence_bonus: ベース報酬 + スコア増加ごとにボーナス

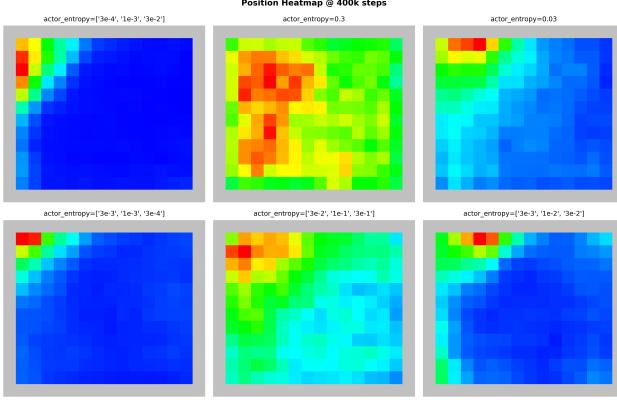


図 10: エントロピー調整実験における探索位置のヒートマップの時間的变化。

- decaying: 時間経過とともに減衰する報酬
- sparse: 完全なシーケンス完了時ののみ報酬
- progressive_stEEP: 急勾配な指数関数的報酬 (3^{pos})
- dense_guidance: ステップごとの細かい報酬(正しい方向+0.1、間違ったタイル-0.1)
- progress_any: flat と同様、スコア増加ごとに一律+1.0

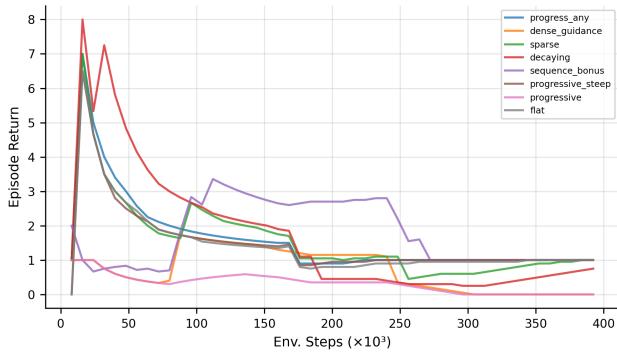


図 11: Pinpad-easy_three タスクにおける異なる報酬設計での学習曲線の比較。

4.6 考察

実験当初は pinpad で学習が進まない理由が報酬が疎であるからだと考えていたが、Pinpad の報酬設計を密にした時にも学習が進まなかったため、そうでは無いと考えた。さらに Atari のモデルの可視化もする中で、Atari でも有意義な方策が学習できていないと思われ、Hieros のモデルそのものに、不安定性あるいは改良の余地があるのではないかと考える。一方で、有意義だと思えない学習をしているにも関わらず、なぜ Hieros が Atari で性能が高く出ていたのかの理由の解明も今後の課題である。Atari の最適方策が学習しやすく、[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とも考えられるが、検証が必要である。検証結果からは Hieros の上手く学習ができるおらず汎化性能が低く、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。

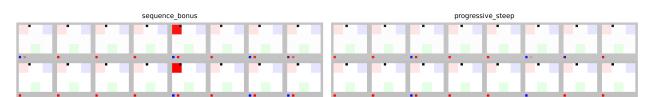
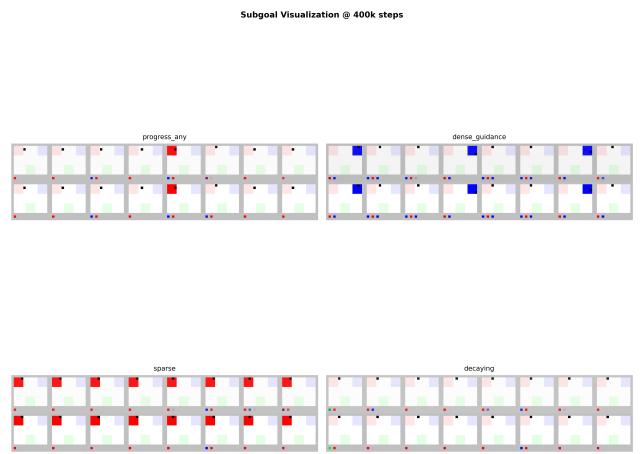


図 12: 報酬設計実験におけるサブゴール可視化の時間的变化。

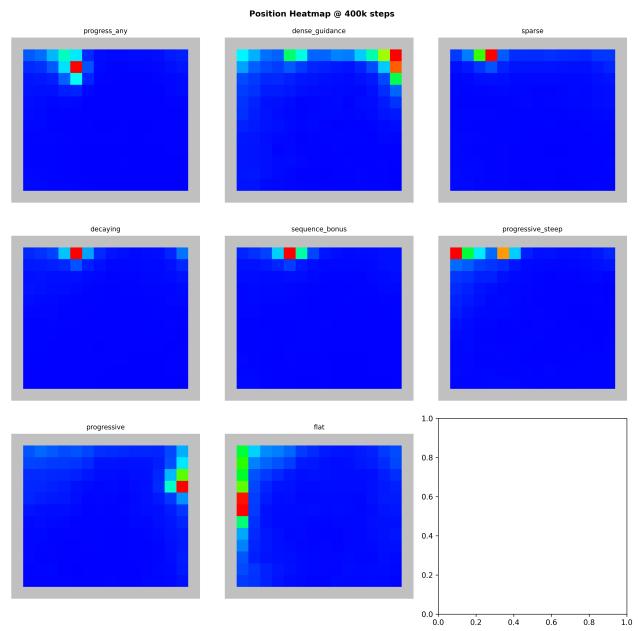


図 13: 報酬設計実験における探索位置のヒートマップの時間的变化。

ベースラインコードにエラーが含まれている可能性は否めない一方で、世界モデルを階層的にすることによって学習が不安定になっている可能性も多い。特に世界モデルは目的関数や誤差蓄積に起因して、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ってハイパーパラメータに繊細であることが知られており、それらの組み合わせによって学習が不安定になってしまっている可能性もある。この理論的、実証実験的解析は今後の課題である。

5.まとめ

6.展望

本稿では Hieros の 3 つの報酬係数を手動で設定し比較を行ったが、HarmonyDream [HarmonyDream] のように、報酬を自動でバランスすると、より多様なタスクにおいて学習ができるようになると予想する。

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] Gumbsch, 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] Patera, S. et al.: Hierarchical Reinforcement Learning: A Comprehensive Survey, ACM Computing Surveys (2021).

7. Appendix

7.1 Atari

Hieros の Atari の実験を再現し、モデルの学習を確認した。スコアは確かに論文通り高かった一方で、policy-image を確認すると単純な方策が学習されており (freeway ではただ前に進む) Hieros では複雑な方策を学べていないのではないかと考える。なお計算資源の都合上、freeway のみは Hieros の元の論文と同じで (batch size, batch length) が (16, 64)、他は (16, 16) で検証を行った。

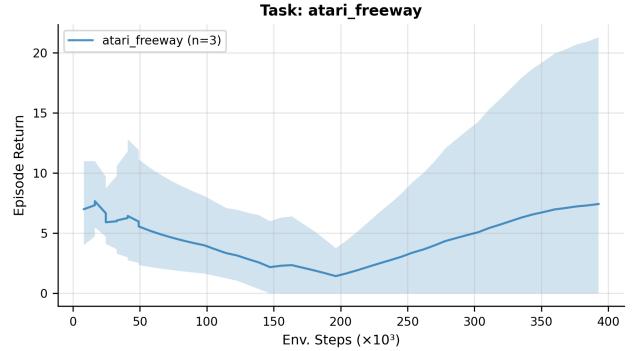


図 14: Atari Freeway タスクの学習曲線 (seed で平均)。

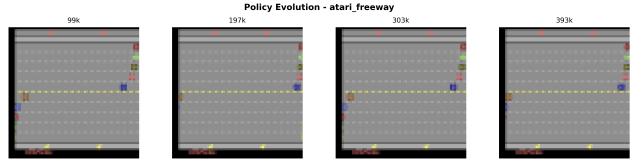


図 15: Atari Freeway タスクにおける方策の時間的变化。

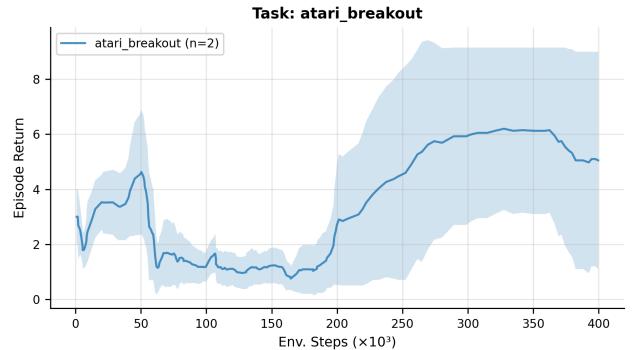


図 16: Atari Breakout タスクの学習曲線 (seed で平均)。

7.2 Director

Director を用いた Pinpad-3 および Pinpad-easy-3 の結果。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 のメモリの都合上途中までしか学習ができなかった。

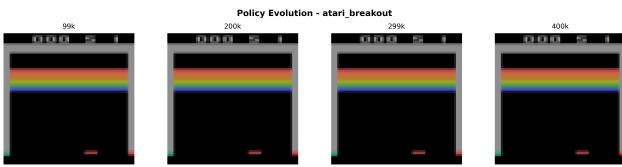


図 17: Atari Breakout タスクにおける方策の時間的变化。

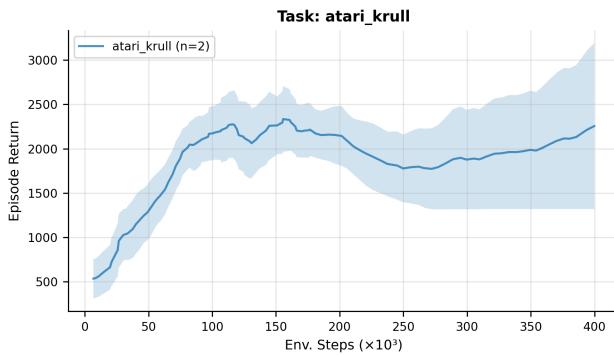


図 18: Atari Krull タスクの学習曲線 (seed で平均)。

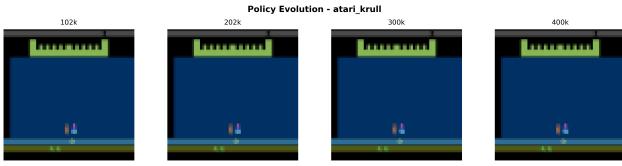


図 19: Atari Krull タスクにおける方策の時間的变化。

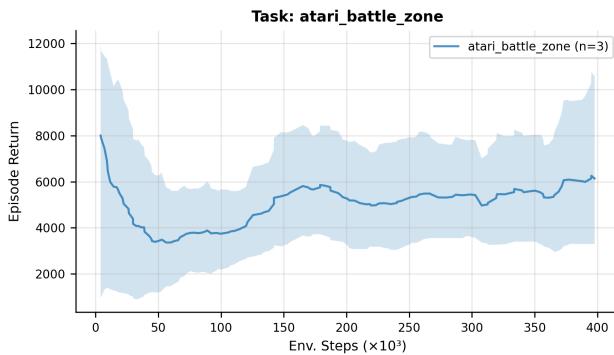


図 20: Atari Battle Zone タスクの学習曲線 (seed で平均)。

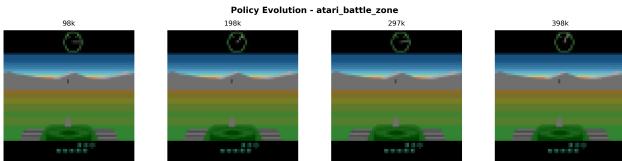


図 21: Atari Battle Zone タスクにおける方策の時間的变化。

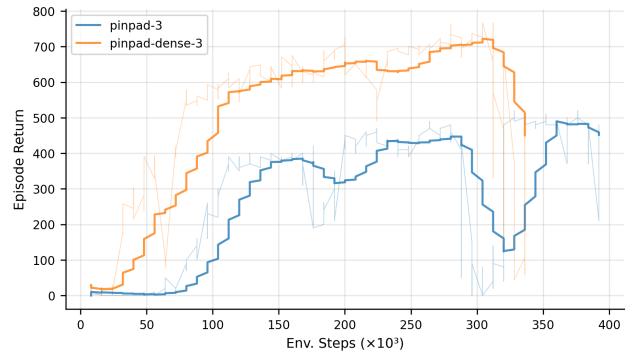


図 22: Director による Pinpad-3 および Pinpad-dense-3 の学習曲線。

7.3 RSSM

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。以下は、RSSM を用いて Hieros を pinpad で結果。

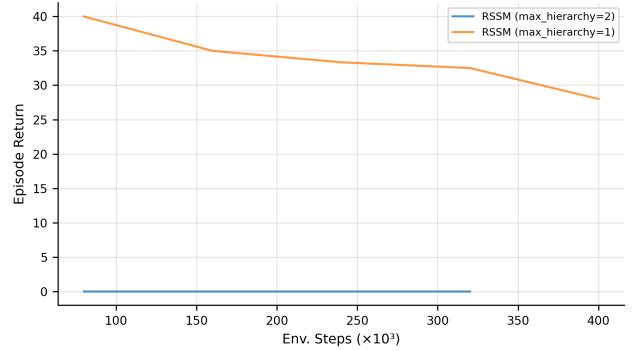


図 23: RSSM を用いた Hieros の Pinpad での学習曲線。

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。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推測できる。