

HRM モデルの Sudoku-Bench への適用

唐澤 嶺¹ 市川 佳彦¹¹ 株式会社 Insight Edge

{rei.karasawa,yoshihiko.ichikawa}@insightedge.jp

概要

本研究は、ニューラルネットワークによる推論能力の特性を検証するため、階層的推論モデル (HRM) と大規模言語モデル (LLM) の比較評価を行った。HRM [1] は、脳の階層構造に着想を得たりカレント型モデルであり、次世代アーキテクチャとして期待されている。本研究では、この HRM の実力を多角的に検証するため、Sudoku-Bench (nikoli_100) [2] を用いた評価を行った。結果、最新の LLM が低迷した一方で、HRM は高い正答率を達成した。

1 はじめに

推論 (Reasoning) は、AI が複雑なタスクを遂行するための核心的な能力である。近年、Chain-of-Thought [3] 等により LLM の推論能力は向上したが、数独のような厳密な制約充足問題はいまだに苦手とされている。コード実行等の外部ツールを用いれば解決自体は可能だが、モデルがどこまで論理的推論を内在的に実行できるかは重要な研究課題である [4, 5]。

また、自然言語処理の文脈では、LLM が推論過程や出力に生じた誤りを外部フィードバックなしに内在的に検出・修正することは依然として難しいことが報告されている [6]。数独は、曖昧さのない制約充足タスクであり、AI モデルの推論能力を診断するベンチマークとして理想的である [2]。

従来の推論モデル評価では、Kaggle Dataset [7, 8] 等の自動生成パズルが用いられてきた。しかし、これらは単純な消去法的操作のみで解くことが可能であり、高度な推論能力を測るには不十分であった [1]。

この課題に対し、Wang らは「計算的難易度 (Computational Difficulty)」に着目し、数独ソルバーのバックトラック数を基準に難易度を高めた Sudoku-Extreme を構築した [1, 9]。一方、Seely らは、人間が作成したパズルに見られる「創造的推

論 (Creative Reasoning)」や「論理的な突破口 (Logical Breakthrough/Break-in)」の重要性を指摘し、これを評価するための Sudoku-Bench を提案している [2]。

本研究では、この Sudoku-Bench を主たるテストベッドとして採用し、Sudoku-Extreme で学習された HRM が、人間的な論理的難易度を要する問題に対して有効かを検証する。同時に、汎用 LLM との比較を通じ、アーキテクチャの違いが推論能力に与える影響を明らかにする。

本稿の貢献は以下の通りである。

- HRM を独立した立場で検証し、提案論文で報告された精度 (約 55%) が再現可能であることを確認した上で、未知のデータに対する評価を行った。
- HRM が、Sudoku-Bench (nikoli_100) に対して 98% という極めて高い性能を持つことを実証した。
- 最新の LLM であっても、ツールを用いない純粋な推論では低精度 (0-4%) となることを示した。

2 階層的推論モデル (HRM)

本研究で評価対象とする Hierarchical Reasoning Model (HRM) [1] は、脳の階層的情報処理と多重時間スケール (Multi-timescale) のメカニズムに着想を得たりカレント型ニューラルネットワークである。

2.1 アーキテクチャ

HRM は、異なる時間スケールで動作する 2 つのりカレントモジュール (Low-level: f_L , High-level: f_H) により構成される。各モジュールは Transformer の Encoder ブロックで実装されており、潜在状態 z を保持する。

入力に対する 1 回の推論プロセスは、 N 回のハイレベル・サイクルと、各サイクル内の T 回のローレベル・ステップによって構成される。全タイムス

ステップを $i = 1, \dots, N \times T$ とするとき、Low-level モジュールの状態更新は以下のように定式化される。

$$z_L^i = f_L(z_L^{i-1}, z_H^{i-1}, \tilde{x}; \theta_L) \quad (1)$$

ここで、 z_L^{i-1} は自身の前の状態、 z_H^{i-1} は現在の High-level 状態（サイクル内では固定）、 \tilde{x} は入力表現である。一方、High-level モジュールは、 T ステップごとに Low-level モジュールの更新結果を受け取り、自身の状態を更新する¹⁾。

$$z_H^i = \begin{cases} f_H(z_H^{i-1}, z_L^i; \theta_H) & \text{if } i \equiv 0 \pmod{T} \\ z_H^{i-1} & \text{otherwise} \end{cases} \quad (2)$$

最後に、全 N サイクル（計 NT ステップ）終了後の High-level 状態 z_H^{NT} から、出力層 f_o を介して最終的な予測 \hat{y} が生成される。

$$\hat{y} = f_o(z_H^{NT}; \theta_o) \quad (3)$$

この階層構造により、HRM はトークンを生成することなく、潜在空間内で情報の更新と保持を繰り返す。

なお、HRM の提案後に、その階層構造を単一のネットワークに簡素化した Tiny Recursive Model (TRM) [10] も提案されている。TRM は Sudoku-Extreme において HRM を上回る性能 (87.4%) が報告されているが、基本的な「潜在空間での再帰的推論」という動作原理は HRM と共通している。本研究では、階層的推論のベースラインとして確立されつつある HRM を評価対象とする。

2.2 学習メカニズムと推論設定

HRM には、効率的に学習を行うための工夫が凝らされている。通常、学習には Backpropagation Through Time (BPTT) が用いられるが、メモリコストが過大となるため、HRM では各モジュールの勾配計算を直近のステップに限定する「One-step Gradient Approximation」を採用している。さらに、順伝播を複数のセグメント（最大 16 回）に分割し、各セグメントの終了時に損失計算と状態の切り離しを行う「Deep Supervision」を導入している。

1) 元論文の式 (2) では f_H の入力が (z_H^{i-1}, z_L^{i-1}) と記載されているが、公式実装およびリポジトリ内での議論 (Issue #24) の通り、記述上の不整合であると考えられる。本稿では、本来の設計意図を反映した (z_H^{i-1}, z_L^i) に修正した。

また、学習時にはモデルが自律的に思考の終了タイミングを決定する「Adaptive Computational Time (ACT)」機構が導入され、効率的な内部表現の獲得が促される。ただし、本研究の評価フェーズ（推論時）においては、元論文の実験設定に倣い、動的な停止は行わず、固定の推論ステップ数（16 ステップ）まで計算を継続させる設定とした。これは、モデルの潜在的な計算能力を最大限に引き出すためである。

3 実験設定

3.1 データセット

本研究では、以下 2 つのデータセットを使用する。

1. **Sudoku-Extreme**: HRM の提案論文において、学習および評価に使用されたデータセットであり、Kaggle, 17-clue, Magictour, Forum-Hard/Extreme といった複数の難易度ソースから収集・統合されたものである。数独ソルバーにとっても探索負荷が高い問題群であることが報告されている [1, 9]。
2. **Sudoku-Bench (nikoli_100)**: Seely ら [2] が提案したベンチマーク。nikoli_100, challenge_100 および ctc という 3 つのサブセットが公開されている。nikoli_100 は、Nikoli 社が Sudoku-Bench のために提供した手作り数独 100 問で構成され、人間的な論理的推論能力の評価を重視して作問されている。challenge_100 は数独バリエーション (Sudoku Variants) 80 問と、nikoli_100 から抽出した 20 問から構成されるベンチマークである。数独バリエーションは、自然言語で記述された追加制約の理解が要求され、言語理解と推論の複合タスクであるといえる。本研究では、言語モデルが複雑なルールを解釈する前段階として、推論性能を評価するため、標準的な数独ルールに基づく nikoli_100 を採用する。

3.2 データセットの静的特徴と重複検証

nikoli_100 と Sudoku-Extreme は、盤面の静的な特徴においては類似している。表 1 に、両データセットの初期状態における空欄数および数字分布の均等性（標準偏差）の比較を示す。

表 1 に示す通り、平均空欄数は約 56 箇所ではほぼ一致しており、数字のばらつきも大きな差はな

表1 データセットの静的難易度指標の比較

指標	nikoli_100	Sudoku-Extreme
空欄数 (平均)	55.72	55.79
空欄数 (中央値)	56.00	56.00
均等性 (標準偏差)	0.689	1.095

い。また、本研究では nikoli_100 が HRM にとって純粋な未知データであることを保証するため、Sudoku-Extreme (約 383 万問) との重複検証を行った。盤面文字列の比較を行い、nikoli_100 の問題は Sudoku-Extreme には含まれていないことを確認した。したがって、nikoli_100 は HRM にとって未知のテストデータである。

3.3 比較モデル

Hierarchical Reasoning Model (HRM) Hugging Face にて公開されている学習済みチェックポイント HRM-checkpoint-sudoku-extreme (1000 examples 版)²⁾ を使用した。学習時と同じく、入力された盤面を潜在空間で処理し、解答を出力させた。

Large Language Models (LLMs) Claude Sonnet 4, Claude Sonnet 4.5, および Gemini 2.5 Pro³⁾ を採用した。評価には Sudoku-Bench 公式リポジトリ [2] で提供されているプロンプトを使用し、以下の 2 設定で検証した。

- **Single-shot:** 解答 (81 マスの数字列) を一括で出力させるモード。プロンプトには推論に関する指示を含めず、モデルの自発的な思考に委ねている。
- **Single-step:** 1 ターンにつき 1 つの数字のみを配置させるモード。システムプロンプトで推論を許可した上で、モデルの出力冒頭を事前入力 (Prefill) し、盤面分析から結論に至る段階的な思考プロセスを明示的に与えることで、CoT を促している。

なお、本実験では、コード実行等の外部ツールは使用していない。パラメータについては、temperature を 0.1、最大出力トークン数を 8,192 に設定し、その他のパラメータは API のデフォルト値を採用した。

3.4 評価指標

モデルの性能は、以下 2 つの指標で評価する。

- 2) <https://huggingface.co/sapientinc/HRM-checkpoint-sudoku-extreme>
- 3) Gemini 2.5 Pro は Thinking 機能を有効化した状態で検証を行った。

表2 Sudoku-Extreme における完全正解率の比較

モデル	完全正解率
本実験	
HRM	55.08%
先行研究 [1] の報告値	
HRM (Original)	55.0%
o3-mini-high	0.0%
Claude 3.7 8K	0.0%
DeepSeek R1	0.0%
Transformer (Direct pred)	0.0%

注：モデル群は元論文 [1] からの引用値であり、本研究では再評価を行っていない点に留意されたい。

1. **完全正解率:** パズル全体の 81 マスがすべて正解と一致した割合。
2. **平均正解配置数:** Single-step 設定において、最初の誤答または停止が発生するまでに、何手目まで正しく数字を配置できたかの平均値 (最大値は空欄数の約 56)。モデルが推論をどこまで維持できたかの指標とする。

4 実験結果

4.1 HRM の再現性検証

本研究の主眼である nikoli_100 での評価に先立ち、HRM が提案論文 [1] と同等の性能を発揮しているかを確認するため、Sudoku-Extreme を用いた再現実験を行った。評価には、提案論文と同じ、全 422,790 問のテストセットを使用した。

表 2 に、本実験での測定結果および元論文での報告値を示す。本実験において、HRM は **55.08%** の完全正解率を記録し、元論文の報告値 (55.0%) を正確に再現した。なお、表 2 下段に示す通り、元論文における最新モデル (o3-mini-high 等) の評価値はいずれも 0.0% であり、本タスクの困難さが示されている。なお、Transformer (Direct pred) は HRM と同様に 8 層 Transformer から構成されるが、階層的な推論の仕組みを持たないモデルである。HRM はわずか 1,000 問の学習データから、約 42 万問のテストデータに対してこれほどの性能を示している。

4.2 Sudoku-Bench における評価

続いて、nikoli_100 に対する評価結果を表 3 に示す。

表3 Sudoku-Bench (nikoli_100) における性能比較

モデル	設定	平均正解配置数	完全正解率
HRM	–	–	98.0%
Claude Sonnet 4	Single-shot	–	0.0%
Claude Sonnet 4	Single-step	2.24	1.0%
Claude Sonnet 4.5	Single-shot	–	0.0%
Claude Sonnet 4.5	Single-step	4.87	4.0%
Gemini 2.5 Pro	Single-shot	–	2.0%
Gemini 2.5 Pro	Single-step	0.60	0.0%

HRM の結果 HRM は、分布外データである nikoli_100 に対して **98.0%** という圧倒的に高い正答率を達成した。Sudoku-Extreme (55.08%) よりも高いスコアが出ている。なお、HRM は入力から潜在空間でのリカレント処理を経て直接解答を出力する仕組みであり、LLM における Single-shot に近いが、内部で複数回の状態更新を内在している点に留意されたい。

LLM の結果 一方、LLM は全モデルにおいて HRM の正解率を大きく下回る結果となった。Single-shot 設定では、Gemini 2.5 Pro のみが正答するケースを確認した。なお、Gemini 2.5 Pro (Single-shot) のみ評価の大半において出力トークンが不足することを確認したため、maxtokens の指定を解除し再検証した。その結果、正解率 2% を得ている。maxtokens を指定した場合の正解率は 0% であった。Single-step 設定においては、Claude Sonnet 4 は 1%、Sonnet 4.5 で 4%、Gemini 2.5 Pro に至っては 0% であった。平均正解配置数を見ても、Sonnet 4.5 で 4.87 手、Gemini 2.5 Pro では 0.60 手と、盤面の序盤で論理的整合性を失っていることがわかる。

5 考察

5.1 HRM の推論挙動の分析

HRM が nikoli_100 に対して 98% の精度を示した要因を解明するため、ACT による内部推論プロセスを分析した。

収束パターンの比較 分析の結果、nikoli_100 で解決した 98 問の平均収束ステップ数は 3.3 (中央値 3) であり、図 1 に示す通りステップ 3 時点で 71%、ステップ 5 時点で 91% のパズルが完全解決に至っていた。比較として、より計算的難易度の高い Sudoku-Extreme からサンプリングした 100 問 (精度 48%) での分析を行ったところ、解決した 48 問の平均収束ステップ数は 5.6 であった。nikoli_100 における早期収束は、同データセットが人間的な「論理

的難易度」は高いものの、HRM にとっては「探索空間の狭い (計算的負荷の低い) 問題」であることを定量的に裏付けている。なお、失敗した 2 問については 16 ステップを通じて正解セル数が 46–63 の範囲で振動する挙動が確認された。

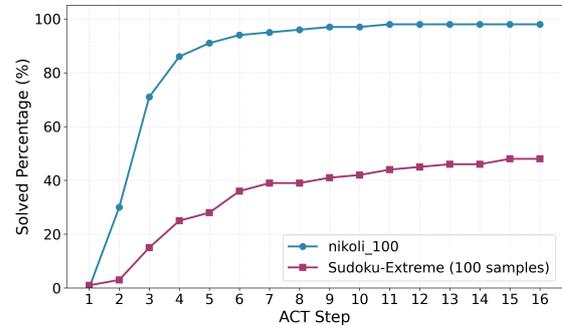


図1 ACT ステップごとの正答率の推移 (nikoli_100 vs Sudoku-Extreme)

5.2 LLM の性能評価における妥当性

複数の最新 LLM (Claude, Gemini) が nikoli_100 を解けなかった結果は、Seely ら [2] の報告とも整合する。彼らのベンチマーク (challenge_100) においても、当時の SOTA モデル (Claude 3.7 Sonnet, GPT-4.1 等) の Single-shot 正答率は 0% であり、推論特化モデルである o3-mini-high を用いても、完全正答率は最大 2.9% (Single-shot, 9×9) に留まっている。

本研究において後継モデルを導入した検証においても、完全正答率は 0–4% という低水準に留まった。この結果は、厳密な制約充足を要する論理パズルの解決が依然として極めて困難な課題であることを示唆している。LLM は数独のルールを暗黙的に学習しているに過ぎないのに対し、HRM は特化型の学習を行なっている。この前提を考慮しても、HRM が未知のデータに対し、人間的な論理の壁を超えて汎化した点は特筆すべきである。

6 おわりに

本研究では、HRM と、最新の LLM 群の推論能力を、nikoli_100 を用いて比較評価した。実験の結果、HRM は精度 98% となった一方、LLM の精度は低迷した (0–4%)。この対照的な結果は、次世代の言語モデルがより高度な推論能力を獲得する上で、HRM のようなアーキテクチャの統合、あるいは HRM 自体の自然言語タスクへの拡張が、有望なアプローチの一つとなり得ることを示唆している。

参考文献

- [1] Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. Hierarchical Reasoning Model. **arXiv preprint arXiv:2506.21734**, 2025.
- [2] Jeffrey Seely, Yuki Imajuku, Tianyu Zhao, Edoardo Cetin, and Llion Jones. Sudoku-Bench: Evaluating creative reasoning with Sudoku variants. **arXiv preprint arXiv:2505.16135**, 2025.
- [3] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In **Advances in Neural Information Processing Systems**, Vol. 35, pp. 24824–24837, 2022.
- [4] Panagiotis Giadikiaroglou, Maria Lymperaiou, Giorgos Filandrianos, and Giorgos Stamou. Puzzle Solving using Reasoning of Large Language Models: A Survey. **arXiv preprint arXiv:2402.11291**, 2024.
- [5] Nemika Tyagi, Mihir Parmar, Mohith Kulkarni, Aswin RRV, Nisarg Patel, Mutsumi Nakamura, Arindam Mitra, and Chitta Baral. Step-by-step Reasoning to Solve Grid Puzzles: Where do LLMs Falter? In **Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing**, pp. 19898–19915, 2024.
- [6] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large Language Models Cannot Self-Correct Reasoning Yet. In **The Twelfth International Conference on Learning Representations**, 2024.
- [7] Kyubyong Park. Can convolutional neural networks crack sudoku puzzles? <https://github.com/Kyubyong/sudoku>, 2018. Accessed: 2025-12-14.
- [8] Kyubyong Park. 1 Million Sudoku Games. <https://www.kaggle.com/datasets/bryanpark/sudoku>, 2018. Accessed: 2025-12-14.
- [9] Sapien Intelligence. Sudoku-Extreme Dataset. <https://huggingface.co/datasets/sapieninc/sudoku-extreme>, 2025. Accessed: 2025-12-14.
- [10] Alexia Jolicoeur-Martineau. Less is More: Recursive Reasoning with Tiny Networks. **arXiv:2510.04871**, 2025.