最近読んだ面白い論文 (AI for Science も含む)

2026/1/07 @slack

---

tax_free

2026/01/07

会の目的

目的

- AI for Science に使えそうな研究を catch up する
- Slack に論文を貼るだけでなく、集まってチームで議論したい
- 朝にやることで生活リズムを整える(朝活)

1 行ルール

- 論文の網羅紹介はたぶんしません。「どうすれば AI for Science に応用できそうか」「どこがおもしろいのか」など個人の視点を中心に話します。

論文 1: Evaluating AI's ability to perform scientific research tasks

A: Contribution

- **この論文の貢献**: 既存のベンチマークでサチっていた/十分な数なかった科学に関する高品質なベンチマークを公開した
- **著者/組織**: OpenAI
- **公開日**: 2025/12/16
- **なぜ今?**: 12/26 の定例で尾崎先生に紹介してもらったのと、科学的な質問に対する回答の評価方法を調べていたから
- **使えるとしたら何ができる?**: 難しい科学的な問題に対する評価方法を参考にして独自のベンチマーク作成に使用できる
- **論文リンク**: https://openai.com/index/frontierscience/

B: Method

1. **対象(入力→出力)**: 専門家が書いた科学タスク(物理/化学/生物)→ モデルの解答 → 正誤/スコア(Olympiad: 正誤、Research: 10 点ルーブリック)
2. **キモ(新規点)**: (1) “サチってない”難問を専門家が新規作成 (2) 研究っぽい open-ended を「チェック可能なルーブリック」に落としてスケール評価(2 トラック構成)
3. **流れ**: 作問(専門家)→ 相互レビュー → 不一致解消 → 改訂で品質担保。Gold set を公開・残りは保持して汚染追跡
4. **評価のしかた**: Olympiad=数式/数値/文字列の同値判定、Research=10 点 rubric で採点(≥7/10 を正解扱い)＋モデル採点(GPT-5 grader)

C-1: Results(フロンティアモデル比較)

Evidence 1

- 条件: FrontierScience-Olympiad(100 問)/ Research(60 問)で複数フロンティアモデルを比較
- 結論: GPT-5.2 がトップ(Olympiad 77% / Research 25%)。Olympiad は Gemini 3 Pro が 76% で僅差
- 意味: 「閉じた難問(Olympiad)はかなり解ける」一方で「研究サブタスク(Research)はまだ余白大」＝評価軸を分けて見る必要あり

C-2: Results(reasoning effort の影響)

Evidence 2

- 条件: reasoning effort(test-time compute / thinking time)を変えて GPT-5.2 と o3 を比較
- 結論: GPT-5.2 は Olympiad 67.5%→77.1%、Research 18%→25%(より長く考えるほど改善)
- 意味: ベンチマーク設計的に、手法差だけでなく推論予算差も明示しないと比較が崩れる

D-1: Gaps(LLM-as-a-judge の揺れ)

Gap 1

- **Applicability Gap**: rubric＋LLM 採点はスケールするが、採点プロンプト依存の揺れや verbosity bias、「自分の出力をよく見積もる」系のバイアスが入りやすい
- **Evidence status**: GPT-5 を model-grader として使い、Research は 10 点 rubric で≥7/10 を正解扱い(=人手スケール不可の現実的解)
- **Minimal test**: 1〜3 問で高品質評価データを作り、モデル名/effort を隠したブラインド採点→採点プロンプトを 2〜3 通りに言い換えてスコア分散を見る

D-2: Gaps(“科学的な取り組み”の評価は未知数)& Next Step

Gap 2

- **Applicability Gap**: 今やってるのは解ける問題。実際には実験しながらデータを集めて検証するプランニングまで評価する必要がある
- **Evidence status**: 制約付きの問題文を解いているので、仮説を作る・計画をたてるという部分は評価できていない
- **Minimal test**: in silico で完結する環境の整備? 自動実験設備との接続?

Next step: まず FrontierScience をそのまま動かして評価用 prompt を少し変えるとどうなるかを確認。良さそうなら rubric や問題を真似して作って評価してみる

論文 2: Gemma Scope 2: helping the AI safety community deepen understanding of complex language model behavior

A: Contribution

- **この論文の貢献**: そこそこ賢い Gemma 3 のモデル内部の解釈をサポートする SAE を作った
- **著者/組織**: Google DeepMind
- **公開日**: 2025/09/16
- **なぜ今?**: 一ヶ月くらい前に出たのと CyberAgent の blog を読んだから
- **使えるとしたら何ができる?**: 質問がどういった特徴量に分解できるのかを定量的に測れる(例: reject された質問がなぜ reject されたのか)
- **論文リンク**: https://deepmind.google/blog/gemma-scope-2-helping-the-ai-safety-community-deepen-understanding-of-complex-language-model-behavior/

B: Method

1. **対象(入力→出力)**: Gemma 3 の各層の activation x → JumpReLU SAE のスパース latent f(x) → 再構成 x^
2. **キモ(新規点)**: JumpReLU+L0 でスパース性を直接制御 / skip connection 付き transcoder で MLP の線形成分を分離
3. **流れ**: データから activation 抽出 → 3 箇所/層に SAE 学習+各層 transcoder 学習 → E2E finetune → 再構成・解釈性・circuit graph で評価
4. **評価のしかた**: reconstruction fidelity (delta LM loss/FVU) / 自動解釈(説明文生成→発火の二値分類) / circuit graph の疎さ

C: Results

Evidence 1 (解釈性の手応え)

- 条件: latent が発火する/しないシーケンスを集め、モデルに「説明」を作らせ、その説明で発火例を当てられるか
- 結論: 低頻度(あまり発火しない)latent ほど解釈しやすい傾向がある
- 意味: AI4S 系の拒否/安全系の挙動を追うなら、“たまにだけ出るトリガー特徴”が見つけやすい可能性

Evidence 2(Transcoder で circuit graph が扱いやすくなる)

- 条件: transcoder/CLT で skip connection あり vs なしを比較(FVU–L0 のトレードオフ、circuit graph の疎さ)
- 結論: skip ありが再構成を改善し、グラフがより疎(少ないノード/エッジで影響を説明)になりやすい
- 意味: 拒否が起きるまでの“経路”を辿るとき、説明が短い回路に圧縮されるとデバッグが現実的になる

D: Gaps & Next Step

Gap 1 (lifescience 領域の“不要な拒否”に直結するか)

- **Applicability Gap**: 論文は一般的な安全・信頼性タスクを念頭。lifescience の“良性質問が拒否される”ケースへの転用は未検証
- **Evidence status**: 「全層 SAE/Transcoder＋評価枠組み」のツール土台提供が主
- **Minimal test**: lifescience 質問を拒否/回答に分けた小セットで発火 latent を集計し、拒否を予測する latent を抽出

Gap 2(“解釈できた”の信頼性)

- **Applicability Gap**: 自動解釈スコアは便利だが、人間にとっての意味解釈やドメイン知識(生物/化学)に対する妥当性は別問題
- **Evidence status**: 解釈性評価は自動解釈(説明生成→二値分類)が中心
- **Minimal test**: 拒否関連 latent 上位 20 個に人手で「何に反応してそうか」ラベルを付け、正当トリガー vs スプリアスに二分して false positive 割合を見積もる

D: Gaps & Next Step

Next step: Gemma Scope 2 の latent を使って「lifescience 質問の拒否を説明する latent catalog」を作り、不要な拒否の典型パターンを可視化する

論文 3: Youtu-Agent: Scaling Agent Productivity with Automated Generation and Hybrid Policy Optimization

A: Contribution

- **この論文の貢献**: context を自動で最適化する training-free の GRPO-like な手法を提案していてソースコードが公開されている (性能改善は怪しい)
- **著者/組織**: Tencent
- **公開日**: 2025/12/26
- **なぜ今?**: Huggingface paper で上位だったから
- **使えるとしたら何ができる?**: ベンチマーク (評価系) さえ用意すれば自動でそれを解くための Agent を作ってくれる
- **論文リンク**: https://arxiv.org/abs/2512.24615

B: Method

1. **対象(入力→出力)**: (1) タスク記述 → agent 設定(YAML)+tool 群(既存 or 自動生成 Python) (2) 少数タスク集合 → “経験メモリ”を context へ注入(Practice) (3) 環境付き長軌道タスク → モデル重み更新(Agent RL)
2. **キモ(新規点)**: 自動生成(tool 検索+不足分は Python tool を合成し YAML 組み立て) / Practice(複数 rollout→LLM 評価で成功/失敗の差分から semantic group advantage を蒸留→context に入れる、重み更新なし)
3. **流れ**: タスク記述 → Agent 自動生成 → 実行 → Practice で並列 rollout+LLM 評価 → 経験テキスト蓄積 → 本番はそれを prompt に注入
4. **評価のしかた**: 生成品質=AgentGen-80 で設定妥当性/ツール実行性/タスク完遂 / Practice=AIME 2024/2025 を Mean@32 / RL=7B で数学・検索系ベンチ前後比較

C-1: Results(General agent 能力:GAIA)

Evidence 1

- 条件: GAIA(466 問)の text-only subset で、pass@1 accuracy を評価(web ツール＋ドキュメント解析＋コード実行を付与)
- 結論: GAIA text-only で 72.8% pass@1
- 意味: 「ツール選択＋多段推論」込みの総合力は一応高い。ただし text-only subset なので、マルチモーダル含む難しさへの外挿は注意

C-2: Results(数学タスク:Practice と RL で“伸びる”が性質が違う)

Evidence 2

- 条件:
  - Practice(training-free GRPO): DAPO-Math-17K から 100 問、3epoch、group size 5。AIME 2024/2025 を Mean@32 で評価
  - Agent RL(重み更新あり): Qwen2.5-7B-Instruct を end-to-end RL、step 500 時点で AIME24/25 を before/after 比較
- 結論:
  - Practice: AIME24 80.0→82.7(+2.7), AIME25 67.9→73.3(+5.4)
  - RL: AIME24 0.10→0.45(+0.35), AIME25 0.09→0.31(+0.22)
- 意味:
  - Practice は少数サンプル＆重み更新なしで改善が出るのが売り(ただし改善幅は控えめ)
  - RL は改善が大きめに見える一方、計算資源・報酬設計・安定化が前提

D-1: Gaps(AI for Science で reward を安定に作れるか)

Gap 1

- **Applicability Gap**: Practice は大量の reward 相当の比較/評価を回す必要がある。AI for Science では正解がない・実験が高コスト・遅いので、「数学ベンチでは回る」≠「科学タスクで回る」
- **Evidence status**: 評価は主に AIME 等(短期・即時採点可能)で、遅延報酬/高コスト評価でスケールできる根拠は薄い
- **Minimal test**: 2〜3 問の小さな科学ミニタスクで意図的に overfit させ、経験メモリが手順・ツール使用・検証ステップをどこまで表現できるか確認

D-2: Gaps(LLM-as-a-Judge の安定性)& Next Step

Gap 2

- **Applicability Gap**: LLM-as-a-Judge で“良い軌跡”を選ぶ設計は、評価者の温度・モデル差・プロンプト差で順位が変わりやすく、学習が不安定に。AI for Science では「どれが良いか」の基準自体が曖昧なので不安定性が増幅しそう
- **Evidence status**: 改善例は示すが、evaluator のブレ耐性(seed/モデル/温度/rubric 変更)を系統的に示していない
- **Minimal test**: 同一ロールアウト集合に対して evaluator を変えて順位相関(Kendall/Spearman)を測る。頻繁に入れ替わるならノイズ駆動リスク

Next step: (i) 2〜3 問 overfit で表現力チェック → (ii) evaluator のブレ定量化 → クリアできたら AI for Science 向けに遅延・高コスト評価の近似設計(proxy reward/段階評価/人手混在)

Appendix

A: Glossary

- **SAE**: Sparse Auto Encoder の略で複雑なモデルを sparse な要素に分解することでモデルの解釈性を高める手法 (LLM に使って解釈性をあげる研究で見る。去年の NeurIPS での評価が高い論文の中に入っていた)
- **GRPO**: Group Relative Policy Optimization の略で、複数の試行結果を比較して良いものを強化学習的に学習する手法 (DeepSeekMath で提案、2025 年にバズった)
- **Gemma**: Google が出している大規模言語モデルシリーズの一つ (https://ai.google.dev/gemma/docs/core?hl=ja) で、Gemini series と違ってオープンソースで提供されている。最近のはそこそこ賢い。

B: Q&A

Q1 (Gemma Scope 2): 一般ユーザーの私たちで reject とかに関してできることあるの?

- A: 基本的には open weight のモデルに限定されるが、意図しない reject だったり、逆にシステムを悪用されないためにも prompt を工夫するなどできることはある。alignment するときでもどういうデータが必要なのか、どれを指標として見ればいいか、などを知るためにも SAE などを用いた explainability は重要。

Q2 (FrontierScience): 評価が低かった回答ってどういうもの?

- A: 私が動かして後で共有します。今週中には共有したい。(TODO)

Q3 (全体): AI4S では prompt と weight のどっちを最適化するのが主流か

- A: Google AI co-scientist、Sakana AI Scientist でも基本的には prompt と tool、agent workflow の最適化で、weight をいじらないことが多い。データが少ない、SoTA モデルの weight を調整するのは時間とお金がかかる、自然言語ほど直感的にフィードバックできないなどが理由。ただし、GPT-5-Codex,

B: Q&A

DeepSeek-Coder など、数学/Coding 系はデータが多く回答を評価しやすいので特化モデルは多い。