DeepSeek-R1:強化学習による大規模言語モデルの推論能力の誘発

DeepSeek-AI

research@deepseek.com

要約

我々は第1世代の推論モデルであるDeepSeek-R1-ZeroとDeepSeek-R1を紹介します。DeepSeek-R1-Zeroは、予備段階として教師あり微調整（SFT）を使用せずに大規模強化学習（RL）を通じて訓練されたモデルであり、優れた推論能力を示しています。RLを通じて、DeepSeek-R1-Zeroは自然に多くの強力で興味深い推論行動を備えています。しかし、可読性の低さや言語混在などの課題に直面しています。これらの問題に対処し、推論性能をさらに向上させるため、我々はRL前のマルチステージトレーニングとコールドスタートデータを含むDeepSeek-R1を導入しています。DeepSeek-R1は推論タスクにおいてOpenAI-o1-1217と同等の性能を達成しています。研究コミュニティをサポートするため、我々はDeepSeek-R1-Zero、DeepSeek-R1、およびDeepSeek-R1から蒸留された6つの密集モデル（1.5B、7B、8B、14B、32B、70B）をQwenとLlamaをベースに公開しています。

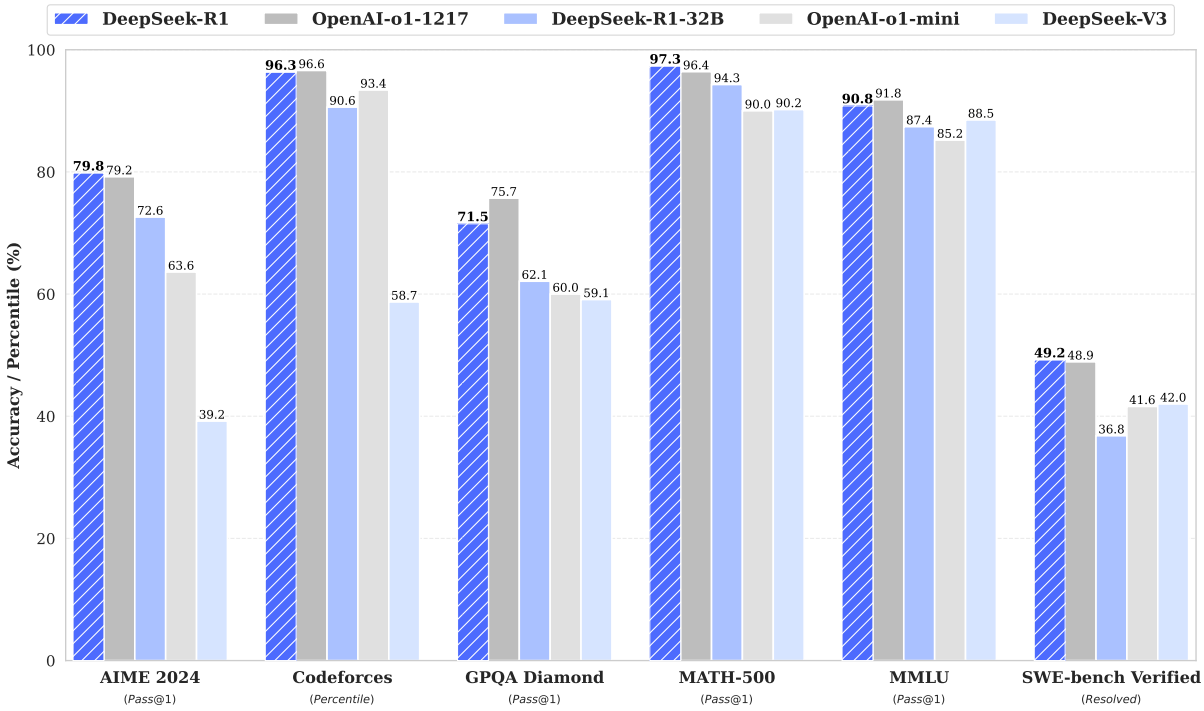

図1 | DeepSeek-R1のベンチマーク性能。

目次

1.はじめに

近年、大規模言語モデル（LLM）は急速に反復と進化を遂行しており（Anthropic、2024；
Google、2024；OpenAI、2024a）、汎用人工知能（AGI）への道を着実に狭めています。
近年、ポストトレーニングは完全なトレーニングパイプラインの重要な構成要素として浮上しています。これは推論タスクの精度向上、社会的価値との整合性、ユーザーの好みへの適応を示すことが証明されており、事前トレーニングに対して比較的少ない計算リソースで済みます。
推論能力の文脈では、OpenAIのo1（OpenAI、2024b）シリーズモデルが推論時スケーリングを導入した最初のモデルであり、思考の連鎖推論プロセスの長さを増加させることで実現しています。このアプローチは数学、コーディング、科学的推論など様々な推論タスクで大きな改善を達成しています。しかし、効果的なテスト時スケーリングの課題は研究コミュニティにとって未解決の問題として残っています。いくつかの先行研究がプロセスベースの報酬モデル（Lightman et al.、2023；Uesato et al.、2022；Wang et al.、2023）、強化学習（Kumar et al.、2024）、モンテカルロ木探索やビーム探索などの探索アルゴリズム（Feng et al.、2024；Trinh et al.、2024；Xin et
al.、2024）を含む様々なアプローチを探索してきました。
しかし、これらのいずれの方法もOpenAIのo1シリーズモデルと比較可能な一般的な推論性能を達成していません。

本論文では、純粋な強化学習（RL）を使用して言語モデル推論能力を改善するための最初のステップを取ります。我々の目標は教師データなしに推論能力を開発するための大規模言語モデルの可能性を探索し、純粋なRLプロセスを通じた自己進化に焦点を当てることです。具体的には、DeepSeek-V3-Baseをベースモデルとして使用し、GRPO（Shao et al.、2024）をRLフレームワークとして採用し、推論においてモデル性能を向上させています。訓練中、DeepSeek-R1-Zeroは自然に多くの強力で興味深い推論行動を備えています。何千ものRLステップの後、DeepSeek-R1-Zeroは推論ベンチマークで優れた性能を示しています。例えば、
AIME 2024のpass@1スコアは15.6%から71.0%に増加し、多数決投票により、スコアはさらに86.7%に向上し、OpenAI-o1-0912の性能と同等です。

しかし、DeepSeek-R1-Zeroは可読性の低さや言語混在などの課題に直面しています。これらの問題に対処し、推論性能をさらに向上させるため、我々はコールドスタートデータとマルチステージトレーニングパイプラインを組み込むDeepSeek-R1を導入しています。具体的には、数千のコールドスタートデータを収集してDeepSeek-V3-Baseモデルを微調整することから始まります。その後、DeepSeek-R1-Zeroと同様の推論指向RLを実行します。RLプロセスでの収束に近づくと、RLチェックポイントに対する棄却サンプリングを通じて新しいSFTデータを作成し、文章作成、事実的質問応答、自己認識などのドメインのDeepSeek-V3からの教師データと組み合わせて、DeepSeek-V3-Baseモデルを再トレーニングします。新しいデータで微調整した後、チェックポイントはすべてのシナリオからのプロンプトを考慮した追加的なRLプロセスを通過し、DeepSeek-R1と呼ばれるチェックポイントが得られ、OpenAI-o1-1217と同等の性能を達成しています。

我々はさらにDeepSeek-R1からより小さな密集モデルへの蒸留を探索しています。ベースモデルとしてQwen2.5-32B（Qwen、2024b）を使用して、DeepSeek-R1からの直接蒸留はそれにRLを適用するよりも優れています。これは、より大きなベースモデルによって発見された推論パターンが推論能力を改善するために重要であることを示しています。我々は蒸留されたQwenおよびLlama（Dubey et al.、2024）シリーズを公開しています。特に、我々の蒸留された14Bモデルは最先端のオープンソースQwQ-32B-Preview（Qwen、2024a）を大幅に上回り、蒸留された32Bおよび70Bモデルは密集モデルの推論ベンチマークで新しい記録を設定しています。

1.1.貢献

ポストトレーニング：ベースモデルへの大規模強化学習

予備段階として教師あり微調整（SFT）に依存することなく、ベースモデルに直接RLを適用しています。このアプローチにより、モデルは複雑な問題を解くための思考の連鎖（CoT）を探索でき、DeepSeek-R1-Zeroの開発をもたらします。DeepSeek-R1-Zeroは自己検証、反省、長い思考の連鎖生成などの能力を示し、研究コミュニティにとって重要なマイルストーンを表しています。特に、SFTの必要なしにRLのみを通じて大規模言語モデルの推論能力を奨励できることを検証する最初のオープン研究です。このブレークスルーは、この分野での将来の進歩への道を開きます。

我々はDeepSeek-R1を開発するパイプラインを導入しています。パイプラインは改善された推論パターンを発見し、人間の好みと整合させることを目指した2つのRL段階、およびモデルの推論能力と非推論能力のシードとして機能する2つのSFT段階を組み込んでいます。我々はこのパイプラインがより良いモデルを作成することで業界に利益をもたらすと信じています。

蒸留：小さいモデルも強力になることができます

より大きなモデルの推論パターンがより小さなモデルに蒸留でき、小さなモデルでのRL発見された推論パターンと比較してより優れた性能を得られることを示しています。オープンソースDeepSeek-R1およびそのAPIは、研究コミュニティが将来的により優れた小さいモデルを蒸留するのに役立つでしょう。

DeepSeek-R1によって生成された推論データを使用して、我々は研究コミュニティで広く使用されている複数の密集モデルを微調整しました。評価結果は蒸留された小さな密集モデルがベンチマークで例外的に良好な性能を示すことを実証しています。DeepSeek-R1-Distill-Qwen-7BはAIME 2024で55.5%を達成し、QwQ-32B-Previewを上回っています。さらに、DeepSeek-R1-Distill-Qwen-32BはAIME 2024で72.6%、MATH-500で94.3%、LiveCodeBenchで57.2%のスコアを獲得しています。これらの結果は以前のオープンソースモデルを大幅に上回り、
o1-miniと比較可能です。我々は1.5B、7B、8B、14B、32B、70Bの蒸留されたチェックポイントをQwen2.5およびLlama3シリーズをベースにコミュニティに公開しています。

1.2.評価結果のサマリー

・推論タスク：（1）DeepSeek-R1はAIME 2024でpass@1スコアの79.8%を達成し、OpenAI-o1-1217をわずかに上回っています。MATH-500では、97.3%という優れたスコアを達成し、OpenAI-o1-1217と同等の性能を示し、他のモデルを大幅に上回っています。（2）コーディング関連タスクでは、DeepSeek-R1はコード競技タスクで専門家レベルを示し、Codeforces上で2,029イロレーティングを達成し、競技の96.3%の人間参加者を上回っています。エンジニアリング関連タスクでは、DeepSeek-R1はDeepSeek-V3よりわずかに良い性能を示し、これは開発者が実世界のタスクを支援することができます。

・知識：MMLU、MMLU-Pro、GPQA Diamondなどのベンチマークでは、DeepSeek-R1は優れた結果を達成し、MMLU上の90.8%、MMLU-Pro上の84.0%、GPQA Diamond上の71.5%のスコアでDeepSeek-V3を大幅に上回っています。これらのベンチマークでの性能はOpenAI-o1-1217よりわずかに低いですが、DeepSeek-R1は他のクローズドソースモデルを上回り、教育的タスクでの競争力を示しています。事実的なベンチマークSimpleQAでは、DeepSeek-R1はDeepSeek-V3を上回り、事実ベースのクエリを処理する能力を示しています。同様のトレンドがOpenAI-o1がこのベンチマークで4oを上回る場合に観察されます。

・**その他**：DeepSeek-R1は創造的執筆、一般的な質問応答、編集、要約など様々なタスクでも優れており、AlpacaEval 2.0で87.6%の長さ制御付き勝率を、Arena-Hardで92.3%の勝率を達成し、非試験指向クエリをインテリジェントに処理する強い能力を示しています。さらに、DeepSeek-R1は長文脈理解が必要なタスクで優れた性能を示し、長文脈ベンチマークではDeepSeek-V3を大幅に上回っています。

2.アプローチ

2.1.概要

先行研究は大量の教師データにモデル性能向上を大きく依存してきました。本研究では、教師あり微調整（SFT）をコールドスタートとして使用しなくても、大規模強化学習（RL）を通じて推論能力を大幅に改善できることを示しています。さらに、少量のコールドスタートデータを含むことで性能をさらに向上させることができます。以下のセクションでは、（1）
SFTデータなしでベースモデルに直接RLを適用するDeepSeek-R1-Zero、および（2）数千の長い思考の連鎖（CoT）例で微調整されたチェックポイントからRLを適用するDeepSeek-R1、（3）DeepSeek-R1から小さな密集モデルへ推論能力を蒸留することを提示します。

2.2.DeepSeek-R1-Zero：ベースモデルへの強化学習

強化学習は推論タスクで大きな効果を示しており、我々の先行研究（Shao et al.、2024；Wang et al.、2023）で証拠付けられています。しかし、これらの研究は大きく教師データに依存しており、収集が時間集約的です。このセクションでは、**教師データなしで**大規模言語モデルが推論能力を開発する可能性を探索し、純粋な強化学習プロセスを通じた自己進化に焦点を当てています。我々はRLアルゴリズムの簡単な概要から始まり、その後いくつかの興味深い結果の提示に続き、このことがコミュニティに価値のある洞察を提供することを望んでいます。

2.2.1.強化学習アルゴリズム

グループ相対ポリシー最適化RLの訓練コストを節約するため、我々はグループ相対ポリシー最適化（GRPO）（Shao et al.、2024）を採用しており、これはポリシーモデルと同じサイズの批評家モデルを放棄し、グループスコアから基線を推定しています。[illegible]的には、各質問について、GRPOは古い[illegible]ポリシーから出力のグループ、、、をサンプルし、次の目的を最大化してポリシーモデルを最適化します：[illegible]

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$
$$\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}A_i, \text{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1-\varepsilon, 1+\varepsilon\right)A_i\right) - \beta\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right)\right), \quad (1)$$

$$\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right) = \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - 1, \quad (2)$$

[illegible]で、はハイパーパラメータであり、はアドバンテージであり、グループ内の各出力に対応する報酬のグループを使用して計算されます：[illegible]。。{}21[illegible]

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \cdots, r_G\})}{\text{std}(\{r_1, r_2, \cdots, r_G\})}. \quad (3)$$

ユーザーとアシスタント間の会話。ユーザーが質問をし、アシスタントがそれを解きます。アシスタントはまず心の中で推論プロセスを考え、その後ユーザーに答えを提供します。推論プロセスと答えはそれぞれ<think></think>および<answer></answer>タグ内に含まれます。つまり、<think>ここに推論プロセス</think><answer>ここに答え</answer>です。ユーザー：プロンプト。アシスタント：

|表1 DeepSeek-R1-Zeroの訓練テンプレート。プロンプトは訓練中に具体的な推論質問に置き換えられます。

2.2.2.報酬モデリング

報酬は訓練信号のソースであり、RLの最適化方向を決定します。DeepSeek-R1-Zeroを訓練するため、我々は主に2つの報酬タイプで構成されるルールベースの報酬システムを採用しています：

- **精度報酬**：精度報酬モデルは応答が正しいかどうかを評価します。例えば、決定論的な結果を持つ数学問題の場合、モデルは指定されたフォーマット（例えば、ボックス内）で最終的な答えを提供することが要求され、正確性の信頼できるルールベース検証を可能にします。同様に、LeetCode問題では、コンパイラを使用して事前定義されたテストケースに基づくフィードバックを生成できます。
- **フォーマット報酬**：精度報酬モデルに加えて、我々は思考プロセスを'<think>'および'</think>'タグ間に配置するようにモデルに強制するフォーマット報酬モデルを採用しています。

DeepSeek-R1-Zeroの開発でアウトカムまたはプロセスニューラル報酬モデルを適用していません。これは、ニューラル報酬モデルが大規模強化学習プロセスで報酬ハッキングに苦しむ可能性があり、報酬モデルの再トレーニングが追加の訓練リソースを必要とし、トレーニングパイプライン全体を複雑にするからです。

2.2.3.訓練テンプレート

DeepSeek-R1-Zeroを訓練するため、我々はベースモデルが指定された指示に従うように導くシンプルなテンプレートを設計することから始まります。表1に描かれているように、このテンプレートはDeepSeek-R1-Zeroに推論プロセスを最初に生成し、その後最終的な答えを生成することを要求しています。我々は構造フォーマットへの制約を意図的に制限し、反省推論の強制や特定の問題解決戦略の促進など、コンテンツ固有の偏見を避けており、RLプロセス中にモデルの自然な進行を正確に観察できることを確保しています。

2.2.4.DeepSeek-R1-Zeroの性能、自己進化プロセスおよびアハモーメント

DeepSeek-R1-Zeroの性能図2はRLトレーニングプロセス全体を通じたAIME 2024ベンチマーク上のDeepSeek-R1-Zeroの性能軌跡を描いています。図示されているように、DeepSeek-R1-ZeroはRL訓練が進むにつれて安定した一貫した性能向上を示しています。特に、AIME 2024上の平均pass@1スコアは初期値の15.6%から71.0%へとジャンプする大幅な増加を示し、OpenAI-o1-0912に匹敵する性能レベルに達しています。この大幅な改善はRLアルゴリズムが時間とともにモデル性能を最適化する有効性を強調しています。

表2は様々な推論関連ベンチマーク全体でDeepSeek-R1-ZeroとOpenAIのo1-0912モデル間の比較分析を提供しています。調査結果はRLが

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCode Bench	CodeForces
	pass@1	cons@64	pass@1	pass@1	pass@1	rating
OpenAI-o1-mini	63.6	80.0	90.0	60.0	53.8	1820
OpenAI-o1-0912	74.4	83.3	94.8	77.3	63.4	1843
DeepSeek-R1-Zero	71.0	86.7	95.9	73.3	50.0	1444

表2 | 推論関連ベンチマーク上のDeepSeek-R1-ZeroおよびOpenAI o1モデルの比較。

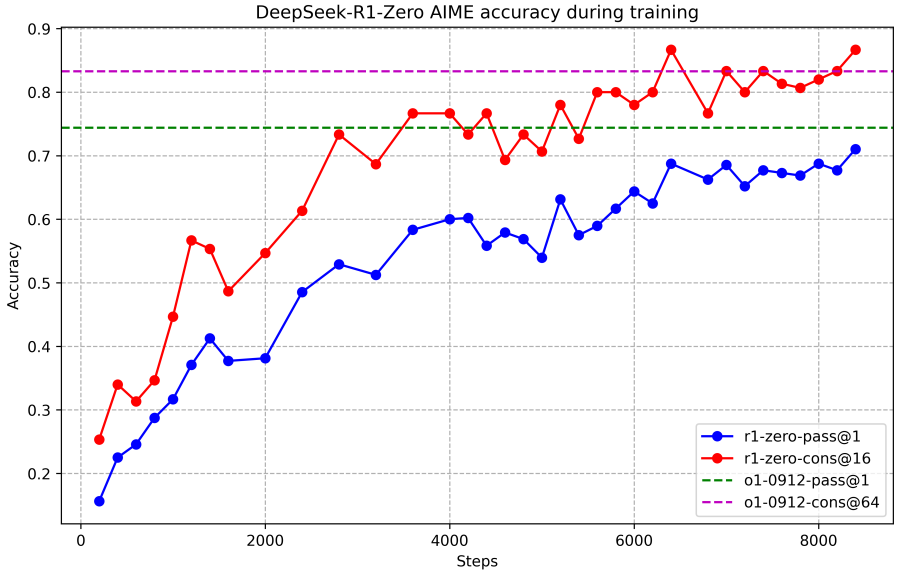

図2 | 訓練中のDeepSeek-R1-ZeroのAIME精度。各質問について、16の応答をサンプリングし、安定した評価を確保するために全体的な平均精度を計算します。

DeepSeek-R1-Zeroが教師あり微調整データの必要なしにRL のみを通じた堅牢な推論能力を獲得することを可能にしています。これは注目に値する成果であり、モデルがRL のみを通じて効果的に学習し、一般化する能力を強調しています。さらに、DeepSeek-R1-Zeroの性能は多数決投票の適用を通じてさらに増強されることができます。例えば、AIEMベンチマークで多数決投票を採用する場合、DeepSeek-R1-Zeroの性能は71.0%から86.7%へと上昇し、OpenAI-o1-0912の性能を超えています。DeepSeek-R1-Zeroが多数決投票の有無にかかわらずそのような競争力のある性能を達成し、非多数決投票を達成する能力は、その強力な基盤能力と推論タスクでの継続的な改善の可能性を強調しています。

DeepSeek-R1-Zeroの自己進化プロセス DeepSeek-R1-Zeroの自己進化プロセスは、RLがモデルの推論能力を自律的に改善するようにどのように駆動できるかの魅力的なデモンストレーションです。ベースモデルから直接RLを開始することで、教師あり微調整段階の影響なしにモデルの進行を密接に監視できます。このアプローチは、特に複雑な推論タスクを処理する能力の観点から、モデルがどのように時間とともに進化するかの明確な見方を提供します。

図3に描かれているように、DeepSeek-R1-Zeroの思考時間はRL プロセス全体を通じた一貫した改善を示し、

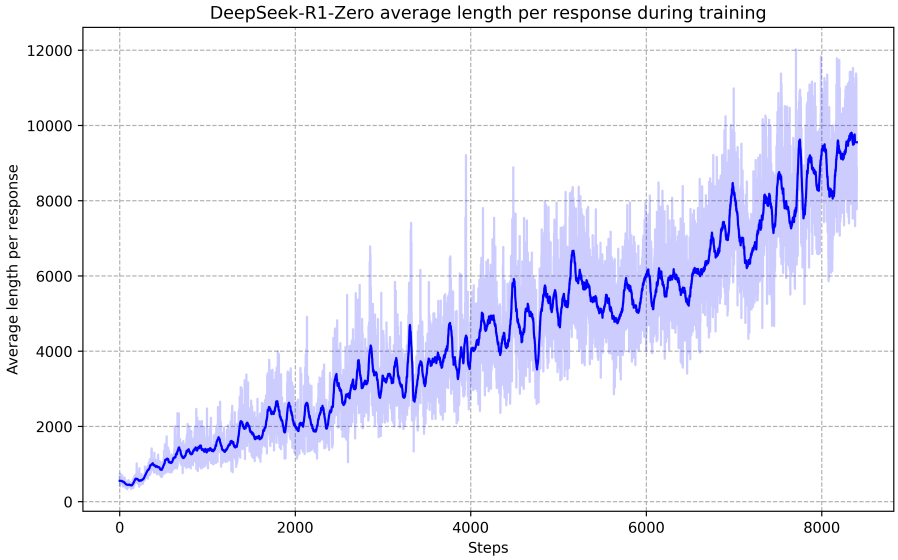

図3 | RLプロセス中の訓練セット上のDeepSeek-R1-Zeroの平均応答長。DeepSeek-R1-Zeroは自然に推論タスクをより多くの思考時間で解くことを学習します。

これは外部調整の結果ではなく、むしろモデル内の固有の開発です。DeepSeek-R1-Zeroは拡張テスト時計算を活用することで、ますます複雑な推論タスクを解く能力を自然に習得します。
この計算は数百から数千の推論トークン生成の範囲であり、モデルはより大きな深さで思考プロセスを探索し精密化することを可能にします。

この自己進化の最も注目すべき側面の1つは、テスト時計算が増加するにつれて洗練された行動が出現することです。モデルが前の段階を再訪し再評価する反省、および問題解決へのアプローチの探索などの行動が自発的に出現します。これらの行動は明示的にプログラムされていませんが、代わりに強化学習環境とのモデルの相互作用の結果として出現しています。この自発的な開発はDeepSeek-R1-Zeroの推論能力を大幅に向上させ、より困難なタスクをより効率的で正確に処理することを可能にします。

DeepSeek-R1-Zeroのアハモーメント DeepSeek-R1-Zeroの訓練中に観察される特に興味深い現象は「アハモーメント」の発生です。この瞬間は、表3に図示されているように、モデルの中間バージョンで発生します。この段階では、DeepSeek-R1-Zeroは初期のアプローチを再評価することで問題により多くの思考時間を配分することを学習します。この行動はモデルの成長する推論能力への証拠であるだけでなく、強化学習がいかに予期しない洗練された結果をもたらすことができるかの魅力的な例でもあります。

この瞬間はモデルにとって単なる「アハモーメント」であるだけでなく、その行動を観察する研究者にとってもアハモーメントです。これは強化学習の力と美しさを強調しています：モデルに問題の解き方を明示的に教える代わりに、我々は単に正しいインセンティブを提供し、それは自律的に高度な問題解決戦略を開発しています。「アハモーメント」は、RLが人工システムに新しいレベルの知性をもたらす可能性を強力に思い出させ、将来的にはより自律的で適応的なモデルへの道を開いています。

√√□□>□□□□質問：1の場合、の実数解の和は質問に等しいですか？応答：
<think>√√□□□□□□..=方程式を解くため、両方を√√2□√□□□□□□□□□□□+===で
乗算することで始めます。内側の平方根の項を分離するために再配置します：□□□□□□□□
□□□□□□□□□□□□□□□□22(−)+⇒−−+()+⇒−−−+(−)=====22 0...ちょっと待って。待ってください。そはここにフラグを立てることができるアハモーメントです。…正しい合計が計算できるかどうかを識別するために、この段階的に再評価しましょう、我々は方程式から始まりました：√√□□□□□□□最初に両方を乗算します：□□□□□□□□
22−+⇒+÷====……次に、両方を乗算することができます、方程式を処理：。。。

表3 | DeepSeek-R1-Zeroの興味深い「アハモーメント」。モデルは擬人化されたトーンを使用して再考することを学習します。これはまた我々にとってのアハモーメントでもあり、強化学習の力と美しさを目撃することを可能にします。

DeepSeek-R1-Zeroの弱点 DeepSeek-R1-Zeroが強力な推論能力を示し、自律的に予期しない強力な推論行動を開発するにもかかわらず、それはいくつかの問題に直面しています。例えば、DeepSeek-R1-Zeroは可読性の低さや言語混在などの課題に苦しんでいます。推論プロセスをより可読にし、オープンコミュニティと共有するため、我々は人間フレンドリーなコールドスタートデータでRLを利用する方法であるDeepSeek-R1を探索しています。

2.3.DeepSeek-R1：コールドスタート付き強化学習

DeepSeek-R1-Zeroの有望な結果に触発されて、2つの自然な質問が生じます：1）少量の高品質データをコールドスタートとして組み込むことで推論性能をさらに向上させたり、収束を加速したりできますか？2）明確で首尾一貫した思考の連鎖（CoT）を生成するだけでなく、強力な一般的能力を示す、ユーザーフレンドリーなモデルをどのようにトレーニングできますか？これらの質問に対処するため、我々はDeepSeek-R1を訓練するためのパイプラインを設計しています。パイプラインは4つの段階で構成されており、以下に概説されています。

2.3.1.コールドスタート

DeepSeek-R1-Zeroとは異なり、ベースモデルからのRL訓練の初期の不安定なコールドスタート段階を防ぐため、DeepSeek-R1では少量の長いCoTデータを構築および収集して、モデルを最初のRLアクターとして微調整しています。そのようなデータを収集するため、長いCoTを例として持つフューショットプロンプトの使用、反省と検証で詳細な答えを生成するようにモデルに直接プロンプトを送る、読可能なフォーマットでDeepSeek-R1-Zero出力を収集する、および人間アノテータによるポストプロセッシングを通じた結果の精密化を含む複数のアプローチを探索してきました。

本研究では、RLの出発点としてDeepSeek-V3-Baseを微調整するため、数千のコールドスタートデータを収集しています。DeepSeek-R1-Zeroと比較して、コールドスタートデータの利点は

次のものを含
みます：

可読性：DeepSeek-R1-Zeroの主な制限は、そのコンテンツがしばしば読むのに適さないということです。応答は複数の言語を混ぜたり、ユーザーの答えを強調するためのマークダウンフォーマットが不足したりする可能性があります。対照的に、DeepSeek-R1のコールドスタートデータを作成する際に、各応答の終わりに要約を含む読みやすいパターンを設計し、読み取りに適さない応答を除外しています。ここで、出力フォーマットを|special_token|<推論プロセス>|special_token|<要約>として定義しています。ここで
、
推論プロセスはクエリのCoTであり、要約は推論結果の要約に使用されます。

可能性：人間の事前知識でコールドスタートデータのパターンを注意深く設計することで、DeepSeek-R1-Zeroに対してより優れた性能を観察しています。我々は反復的トレーニングが推論モデルの方法であるとより良い方法であると信じています。

2.3.2.推論指向の強化学習

コールドスタートデータ上のDeepSeek-V3-Baseの微調整後、我々はDeepSeek-R1-Zeroで採用されたのと同じ大規模強化学習トレーニングプロセスを適用しています。この段階は、コーディング、数学、科学、論理推論などの推論集約的なタスク、明確に定義された問題を含む、明確な解決策を持つものにおいて、特にモデルの推論能力の強化に焦点を当てています。トレーニングプロセス中に、CoTはしばしば言語混在を示し、特にRLプロンプトが複数の言語を含む場合に表示されます。言語混在の問題を軽減するため、我々はRL訓練中に言語一貫性報酬を導入し、これはCoTにおけるターゲット言語単語の割合として計算されます。アブレーション実験はそのような整合により、モデル性能のわずかな低下が生じることを示していますが、この報酬は人間の好みと整合し、より読みやすくしています。最終的に、推論タスクの精度と言語一貫性の報酬を直接合計することで組み合わせて、最終報酬を形成しています。その後、推論タスクでの収束を達成するまで、微調整されたモデルでRL訓練を適用します。

2.3.3.棄却サンプリングと教師あり微調整

推論指向RLが収束すると、結果のチェックポイントを使用してその後のラウンドのためのSFT（教師あり微調整）データを収集しています。最初のコールドスタートデータは主に推論に焦点を当てていたのとは異なり、この段階は他のドメインからのデータを組み込んで、執筆、ロールプレイ、および他の一般的な目的のタスクでモデルの能力を強化しています。具体的には、以下に説明するようにデータを生成し、モデルを微調整しています。

推論データ推論プロンプトをキュレートし、上記のRLトレーニングからのチェックポイントで棄却サンプリングを実行することにより推論軌跡を生成しています。前の段階では、ルールベース報酬を使用して評価できるデータのみを含めていました。しかし、この段階では、地面の真実とモデル予測をDeepSeek-V3に供給することにより、生成的報酬モデルを使用するいくつかのデータを組み込むことで、データセットを拡張しています。さらに、モデル出力が時々混沌としており、読むのが困難なため、混在言語、長いパラグラフ、コードブロックを含む思考の連鎖を除外しています。各プロンプトについて、複数の応答をサンプリングし、正しいもののみを保持しています。合計で、約600kの推論関連訓練サンプルを収集しました。

非推論データ執筆、事実的質問応答、自己認識、翻訳などの非推論データでは、我々はDeepSeek-V3パイプラインを採用し、DeepSeek-V3のSFTデータセットの一部を再利用しています。特定の非推論タスクでは、プロンプトすることでクエリに答える前に潜在的な思考の連鎖を生成するためにDeepSeek-V3を呼び出しています。しかし、「こんにちは」などのより簡単なクエリでは、応答にCoTを提供していません。最終的に、推論と無関の約200kの訓練サンプルを合計で収集しました。

我々は上記のキュレートされた約800kサンプルのデータセットを使用して、2エポックのDeepSeek-V3-Baseを微調整しています。

2.3.4.すべてのシナリオに対する強化学習

モデルを人間の好みとさらに整合させるため、モデルの有用性と無害性を改善しながら同時に推論能力を改善することを目的とした二次的な強化学習段階を実装しています。具体的には、報酬信号と多様なプロンプト分布の組み合わせを使用してモデルを訓練しています。推論データでは、数学、コード、論理推論ドメインの学習プロセスを導くルールベース報酬を利用するDeepSeek-R1-Zeroで概説された方法論に従っています。一般的なデータでは、複雑で微妙なシナリオでの人間の好みをキャプチャするために報酬モデルに頼っています。我々はDeepSeek-V3パイプラインに基づいており、嗜好ペアと訓練プロンプトの同様の分布を採用しています。有用性のために、最終要約に専念し、評価が応答のユーザーへの効用と関連性を強調しながら、基盤となる推論プロセスへの干渉を最小化することを確保しています。無害性のため、推論プロセスと要約の両方を含むモデルの全応答を評価し、生成プロセス中に発生する可能性がある潜在的なリスク、偏見、有害なコンテンツを識別および軽減します。最終的に、報酬信号と多様なデータ分布の統合により、我々は推論で優れており、有用性と無害性を優先するモデルを訓練できます。

2.4.蒸留：小型モデルに推論能力を付与

DeepSeek-R1のようなより効率的な小型モデルに推論能力を備えるため、我々はQwen（Qwen、2024b）およびLlama（AI@Meta、2024）などのオープンソースモデルを、§2.3.3で詳述された800kサンプルでキュレートされたもので直接微調整しています。我々の調査結果はこのシンプルな蒸留方法が小型モデルの推論能力を大幅に向上させることを示しています。ここで使用するベースモデルはQwen2.5-Math-1.5B、Qwen2.5-Math-7B、Qwen2.5-14B、Qwen2.5-32B、Llama-3.1-8B、およびLlama-3.3-70B-Instructです。Llama-3.3を選択した理由は、その推論能力がLlama-3.1のものよりわずかに優れているからです。

蒸留されたモデルでは、RLを含むRL段階を含めず、SFTのみを適用しています。RL段階を組み込むことはモデル性能を大幅に向上させることができますが、我々の主な目標はここで蒸留技術の有効性を示すことであり、RLステージの探索はより広い研究コミュニティに残しています。

3.実験

ベンチマーク我々はMMUL（Hendrycks et al.、2020）、MMLU-Redux（Gemate al.、2024）、MMLU-Pro（Wang et al.、2024）、C-Eval（Huang et al.、2023）、およびCMMUU（Li et al.、2023）、IFEval（Zhou et al.、2023）、FRAMES（Krishna et al.、2024）、GPQA Diamond（Rein et al.、2023）、SimpleQA（OpenAI、2024c）、C-SimpleQA（He et al.、2024）、SWE-Bench Verified（OpenAI、

2024d）、Aider 1、LiveCodeBench（Jain et al.、2024）（2024-08 - 2025-01）、Codeforces 2、中国国立高校数学オリンピック（CNMO 2024）3、およびアメリカ招待数学試験2024（AIME 2024）（MAA、2024）。標準的なベンチマークに加えて、我々は大規模言語モデルを審判として使用するオープンエンド生成タスクでモデルを評価しています。具体的には、ペアワイズ比較のために審判としてGPT-4-Turbo-1106を活用するAlpacaEval 2.0（Dubois et al.、2024）およびArena-Hard（Li et al.、2024）の元の構成に従っています。ここで、長さバイアスを回避するために最終要約のみを評価に供給しています。蒸留されたモデルでは、AIME 2024、MATH-500、GPQA Diamond、Codeforces、およびLiveCodeBenchの代表的な結果を報告しています。

評価プロンプトDeepSeek-V3のセットアップに従い、MMLU、DROP、GPQA Diamond、SimpleQAなどの標準的なベンチマークはsimple-evalsフレームワークのプロンプトを使用して評価されます。MMLU-Reduxについては、ゼロショット設定でZero-Evalプロンプト形式(Lin、2024)を採用しています。MMLU-Pro、C-Eval、CLUE-WSCについては、元のプロンプトがフューショットであるため、プロンプトをゼロショット設定に若干修正しています。フューショットの思考の連鎖はDeepSeek-R1のパフォーマンスを低下させる可能性があります。その他のデータセットは、作成者が提供するデフォルトプロンプトを使用して、元の評価プロトコルに従います。コードおよび数学ベンチマークについては、HumanEval-Mulデータセットは8つの主要なプログラミング言語(Python、Java、C++、C#、JavaScript、TypeScript、PHP、Bash)をカバーしています。LiveCodeBenchでのモデルパフォーマンスは思考の連鎖形式を使用して評価され、データは2024年8月から2025年1月の間に収集されました。Codeforces データセットは10のDiv.2コンテストの問題と専門家が作成したテストケースを使用して評価され、その後、予想されるレーティングと競争者のパーセンテージが計算されます。SWE-Benchの検証済み結果はagentlessフレームワーク(Xia et al.、2024)を介して取得されます。AIDER関連ベンチマークは「diff」形式を使用して測定されます。DeepSeek-R1の出力は、各ベンチマークについて最大32,768トークンに制限されます。

ベースラインDeepSeek-V3、Claude-Sonnet-3.5-1022、GPT-4o-0513、OpenAI-o1-mini、OpenAI-o1-1217を含む複数の強いベースラインに対して包括的な評価を実施しています。中国本土でのOpenAI-o1-1217 APIへのアクセスが困難なため、公式レポートに基づくそのパフォーマンスを報告しています。蒸留されたモデルについては、オープンソースモデルQwQ-32B-Preview(Qwen、2024a)も比較しています。

評価セットアップモデルの最大生成長を32,768トークンに設定しています。長い出力を持つ推論モデルを評価するために貪欲デコーディングを使用すると、繰り返しレートが高くなり、異なるチェックポイント間で大きなばラつきが生じることがわかりました。したがって、□pass@評価(Chen et al.、2021)をデフォルトとし、ゼロでない温度を使用してpass@1を報告しています。□□具体的には、サンプリング温度0.6とtop-p値0.95を使用して、各質問に対して(テストセットサイズに応じて通常は4〜64の間)応答を生成します。Pass@1はその後、□□として計算されます。

$$\text{pass@1} = \frac{1}{k}\sum_{i=1} p_i,$$

□□□で、□番目の応答の正確性を示しています。この方法はより信頼性の高いパフォーマンス推定値を提供します。AIME 2024については、64個のサンプルを使用したコンセンサス(多数決投票)結果(Wang et al.、2022)も報告しており、cons@64で示されています。

1 https://aider.chat

2 https://codeforces.com

3 https://www.cms.org.cn/Home/comp/comp/cid/12.html

3.1.DeepSeek-R1評価

	Benchmark (Metric)	Claude-3.5-Sonnet-1022	GPT-4o 0513	DeepSeek V3	OpenAI o1-mini	OpenAI o1-1217	DeepSeek R1
	Architecture	-	-	MoE	-	-	MoE
	# Activated Params	-	-	37B	-	-	37B
	# Total Params	-	-	671B	-	-	671B
English	MMLU (Pass@1)	88.3	87.2	88.5	85.2	**91.8**	90.8
	MMLU-Redux (EM)	88.9	88.0	89.1	86.7	-	**92.9**
	MMLU-Pro (EM)	78.0	72.6	75.9	80.3	-	**84.0**
	DROP (3-shot F1)	88.3	83.7	91.6	83.9	90.2	**92.2**
	IF-Eval (Prompt Strict)	**86.5**	84.3	86.1	84.8	-	83.3
	GPQA Diamond (Pass@1)	65.0	49.9	59.1	60.0	**75.7**	71.5
	SimpleQA (Correct)	28.4	38.2	24.9	7.0	**47.0**	30.1
	FRAMES (Acc.)	72.5	80.5	73.3	76.9	-	**82.5**
	AlpacaEval2.0 (LC-winrate)	52.0	51.1	70.0	57.8	-	**87.6**
	ArenaHard (GPT-4-1106)	85.2	80.4	85.5	92.0	-	**92.3**
Code	LiveCodeBench (Pass@1-COT)	38.9	32.9	36.2	53.8	63.4	**65.9**
	Codeforces (Percentile)	20.3	23.6	58.7	93.4	**96.6**	96.3
	Codeforces (Rating)	717	759	1134	1820	**2061**	2029
	SWE Verified (Resolved)	**50.8**	38.8	42.0	41.6	48.9	49.2
	Aider-Polyglot (Acc.)	45.3	16.0	49.6	32.9	**61.7**	53.3
Math	AIME 2024 (Pass@1)	16.0	9.3	39.2	63.6	79.2	**79.8**
	MATH-500 (Pass@1)	78.3	74.6	90.2	90.0	96.4	**97.3**
	CNMO 2024 (Pass@1)	13.1	10.8	43.2	67.6	-	**78.8**
Chinese	CLUEWSC (EM)	85.4	87.9	90.9	89.9	-	**92.8**
	C-Eval (EM)	76.7	76.0	86.5	68.9	-	**91.8**
	C-SimpleQA (Correct)	55.4	58.7	**68.0**	40.3	-	63.7

表4 | DeepSeek-R1と他の代表的なモデルの比較。

MMLU、MMLU-Pro、GPQA Diamondなどの教育志向の知識ベンチマークについて、DeepSeek-R1はDeepSeek-V3と比較して優れたパフォーマンスを示しています。この改善は主に、大規模な強化学習を通じて大きな成果が達成されるSTEM関連の質問の精度が向上していることに起因しています。さらに、DeepSeek-R1はFRAMES(長文脈依存の質問応答タスク)で優れており、強力なドキュメント分析能力を示しています。これはAI駆動の検索およびデータ分析タスクにおける推論モデルの可能性を強調しています。事実的なベンチマークSimpleQAでは、DeepSeek-R1がDeepSeek-V3を上回り、事実ベースのクエリを処理する能力を示しています。OpenAI-o1がこのベンチマークでGPT-4oを上回る同様の傾向が観察されています。ただし、DeepSeek-R1は中国語SimpleQAベンチマークではDeepSeek-V3より性能が低くなっています。これは主にセーフティ強化学習後に特定のクエリへの回答を拒否する傾向があるためです。セーフティ強化学習がなければ、DeepSeek-R1は70%を超える精度を達成できました。

DeepSeek-R1はまた、モデルのフォーマット指示に従う能力を評価するために設計されたベンチマークであるIF-Evalで印象的な結果を提供しています。これらの改善は、教師あり微調整(SFT)および強化学習トレーニングの最終段階での指示追従データの組み込みに関連しています。さらに、AlpacaEval2.0およびArena-Hardで優れたパフォーマンスが観察され、DeepSeek-R1の執筆タスクおよびオープンドメイン質問応答における強みが示されています。DeepSeek-V3を大幅に上回るパフォーマンスは、推論能力を高めるだけでなく、多様なドメイン全体でパフォーマンスを向上させる大規模な強化学習の汎化上の利点を強調しています。さらに、DeepSeek-R1で生成された要約の長さは簡潔で、Arena-Hardで平均689トークン、AlpacaEval 2.0で2,218文字です。
これは、

DeepSeek-R1がGPTベースの評価中に長さバイアスを導入することを回避し、複数のタスク全体でロバスト性をさらに強化しています。

数学タスクでは、DeepSeek-R1はOpenAI-o1-1217と同等のパフォーマンスを示し、他のモデルを大幅に上回っています。LiveCodeBenchおよびCodeforcesなどのコーディングアルゴリズムタスクでは、推論に焦点を当てたモデルがこれらのベンチマークを支配する同様の傾向が観察されています。エンジニアリング指向のコーディングタスクでは、OpenAI-o1-1217はAiderではDeepSeek-R1を上回りますが、SWE Verifiedではそれに匹敵するパフォーマンスを達成しています。DeepSeek-R1のエンジニアリング性能は次のバージョンで改善されると考えています。現在、関連する強化学習トレーニングデータの量は非常に限定されているためです。

3.2.蒸留モデル評価

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCode Bench	CodeForces
	pass@1	cons@64	pass@1	pass@1	pass@1	rating
GPT-4o-0513	9.3	13.4	74.6	49.9	32.9	759
Claude-3.5-Sonnet-1022	16.0	26.7	78.3	65.0	38.9	717
OpenAI-o1-mini	63.6	80.0	90.0	60.0	53.8	**1820**
QwQ-32B-Preview	50.0	60.0	90.6	54.5	41.9	1316
DeepSeek-R1-Distill-Qwen-1.5B	28.9	52.7	83.9	33.8	16.9	954
DeepSeek-R1-Distill-Qwen-7B	55.5	83.3	92.8	49.1	37.6	1189
DeepSeek-R1-Distill-Qwen-14B	69.7	80.0	93.9	59.1	53.1	1481
DeepSeek-R1-Distill-Qwen-32B	**72.6**	83.3	94.3	62.1	57.2	1691
DeepSeek-R1-Distill-Llama-8B	50.4	80.0	89.1	49.0	39.6	1205
DeepSeek-R1-Distill-Llama-70B	70.0	**86.7**	**94.5**	**65.2**	**57.5**	1633

表5 | DeepSeek-R1蒸留モデルと推論関連ベンチマークの他の同等モデルとの比較。

表5に示すように、DeepSeek-R1の出力を単純に蒸留することで、効率的なDeepSeek-R1-7B(すなわち、DeepSeek-R1-Distill-Qwen-7B、以下同様に略記)は、全面的にGPT-4o-0513などの非推論モデルを上回ることができます。DeepSeek-R1-14Bはすべての評価メトリクスでQwQ-32B-Previewを上回り、DeepSeek-R1-32BおよびDeepSeek-R1-70Bはほとんどのベンチマークでo1-miniを大幅に超えています。これらの結果は蒸留の強い可能性を示しています。さらに、これらの蒸留されたモデルに強化学習を適用すると、さらに大きな改善が得られることがわかりました。これにより、さらなる探索の価値があると考えられるため、ここでは単純な教師あり微調整で蒸留されたモデルの結果のみを提示しています。

4.議論

4.1.蒸留 vs 強化学習

セクション3.2では、DeepSeek-R1を蒸留することで、小さいモデルが印象的な結果を達成できることがわかります。しかし、まだ1つの質問が残っています。モデルは蒸留なしでこの論文で説明されている大規模な強化学習トレーニングを通じて同等のパフォーマンスを達成できるでしょうか?

この質問に答えるために、Qwen-32B-Baseで数学、コード、STEMデータを使用して大規模な強化学習トレーニングを実施し、10,000ステップ以上トレーニングした結果、DeepSeek-R1-Zero-Qwen-32Bが得られました。実験結果は表6に示されており、32Bベースモデルが大規模な

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCodeBench
	pass@1	cons@64	pass@1	pass@1	pass@1
QwQ-32B-Preview	50.0	60.0	90.6	54.5	41.9
DeepSeek-R1-Zero-Qwen-32B	47.0	60.0	91.6	55.0	40.2
DeepSeek-R1-Distill-Qwen-32B	**72.6**	**83.3**	**94.3**	**62.1**	**57.2**

表6 | 蒸留およびRLモデルの推論関連ベンチマークの比較。

強化学習トレーニング後、QwQ-32B-Previewと同等のパフォーマンスを達成しています。ただし、DeepSeek-R1から蒸留されたDeepSeek-R1-Distill-Qwen-32Bは、すべてのベンチマークでDeepSeek-R1-Zero-Qwen-32Bを大幅に上回ります。

したがって、次の2つの結論を引き出すことができます。第1に、より強力なモデルをより小さいモデルに蒸留すると優れた結果が得られますが、この論文で言及されている大規模な強化学習に依存する小さいモデルは膨大な計算能力を必要とし、蒸留のパフォーマンスを達成できない可能性があります。第2に、蒸留戦略は経済的で効果的ですが、知能の境界を超えて進むには、より強力なベースモデルと大規模な強化学習が必要な場合があります。

4.2.失敗した試み

DeepSeek-R1の開発の初期段階では、失敗と挫折に直面しました。これらの失敗の経験をここで共有して洞察を提供しますが、これらのアプローチが効果的な推論モデルの開発に能力がないことを意味しません。

プロセスベースの報酬モデル(PRM)PRMは、モデルが推論タスクを解く際にはより良いアプローチに向かうよう誘導する合理的な方法です(Lightman et al.、2023; Uesato et al.、2022; Wang et al.、2023)。しかし、実際には、PRMには3つの主な制限があり、最終的な成功を阻害する可能性があります。第1に、推論全般で細かい粒度のステップを明示的に定義することは困難です。第2に、現在の中間ステップが正しいかどうかを判断することは困難な作業です。モデルを使用した自動注釈は満足のいく結果をもたらさない可能性があり、手動注釈はスケーリングアップに適していません。第3に、モデルベースのPRMが導入されると、必然的に報酬ハッキングにつながり(Gao et al.、2022)、報酬モデルの再トレーニングは追加のトレーニングリソースを必要とし、全体的なトレーニングパイプラインが複雑になります。結論として、PRMはモデルが生成したトップNの応答を再ランク付けするか、ガイド検索を支援する(Snell et al.、2024)能力を示していますが、その利点は、当社の実験における大規模な強化学習プロセス中に導入される追加の計算オーバーヘッドと比較して限定的です。

モンテカルロ木探索(MCTS)AlphaGo(Silver et al.、2017b)およびAlphaZero(Silver et al.、2017a)に触発され、モンテカルロ木探索(MCTS)を使用してテスト時計算のスケーラビリティを強化することを検討しました。このアプローチは、答えをより小さい部分に分割して、モデルが解決空間を体系的に探索できるようにすることを含みます。これを容易にするために、検索に必要な特定の推論ステップに対応する複数のタグを生成するようにモデルにプロンプトを出します。
トレーニングについては、まず収集されたプロンプトを使用して、事前トレーニングされた価値モデルによってガイドされるMCTSを介して答えを見つけます。その後、結果の質問応答ペアを使用して、アクターモデルと価値モデルの両方をトレーニングし、プロセスを反復的に洗練します。
しかし、このアプローチはトレーニングをスケールアップする際に複数の課題に直面しています。第1に、チェスとは異なり、検索空間が比較的明確に定義されている場合、トークン生成は

指数関数的に大きい検索空間が提示されます。これに対処するために、各ノードの最大拡張制限を設定しましたが、これはモデルがローカル最適値に行き詰まる可能性があります。第2に、価値モデルは検索の各ステップをガイドするため、生成の品質に直接影響します。細かい粒度の価値モデルをトレーニングすることは本質的に困難であり、モデルが反復的に改善することを困難にしています。AlphaGoのコア成功は、価値モデルをトレーニングしてパフォーマンスを段階的に向上させることに依存していましたが、トークン生成の複雑さのため、この原則は当社のセットアップで再現することが困難であることが判明しました。

結論として、MCTSは事前トレーニングされた価値モデルと組み合わせた場合、推論中にパフォーマンスを改善できますが、自己検索を通じてモデルパフォーマンスを反復的に向上させることは依然として大きな課題です。

5.結論、制限、および今後の研究

この研究では、強化学習を通じてモデルの推論能力を向上させるための取り組みをお伝えします。
DeepSeek-R1-Zeroはコールドスタートデータに依存しない純粋な強化学習アプローチを表し、様々なタスク全体で強力なパフォーマンスを達成しています。DeepSeek-R1はより強力で、コールドスタートデータと反復的な強化学習による微調整を活用しています。最終的に、DeepSeek-R1は様々なタスクでOpenAI-o1-1217と同等のパフォーマンスを達成しています。

推論能力を小さい密集モデルに蒸留することをさらに探索しました。DeepSeek-R1をティーチャーモデルとして使用して800,000のトレーニングサンプルを生成し、複数の小さい密集モデルを微調整しました。結果は有望です。DeepSeek-R1-Distill-Qwen-1.5BはGPT-4oおよびClaude-3.5-Sonnetをすべての数学ベンチマークで上回ります。AIIMEで28.9%、MATHで83.9%を達成しています。その他の密集モデルも印象的な結果を達成し、同じ基盤となるチェックポイントに基づく他の指示調整モデルを大幅に上回っています。

今後、DeepSeek-R1の以下の方向での研究に投資する予定です。

一般的な能力:・現在のところ、DeepSeek-R1の能力は関数呼び出し、マルチターン、複雑なロールプレイ、JSONoutputなどのタスクではDeepSeek-V3に及びません。今後、長い思考の連鎖を活用してこれらの分野のタスクを強化する方法を探索する予定です。

言語混在:DeepSeek-R1は現在、中国語と英語に最適化されており、・他の言語でのクエリを処理する際に言語混在の問題が生じる可能性があります。例えば、DeepSeek-R1はクエリが英語や中国語以外の言語にある場合でも、推論と応答に英語を使用する可能性があります。今後のアップデートでこの制限に対処することを目指しています。

プロンプトエンジニアリング:・DeepSeek-R1を評価する際、プロンプトに対する感度が高いことを観察しています。フューショットプロンプティングは一貫してそのパフォーマンスを低下させています。最適な結果を得るには、ゼロショット設定を使用して問題を直接説明し、出力フォーマットを指定することをお勧めします。

ソフトウェアエンジニアリングタスク:長い評価時間により、・強化学習プロセスの効率に影響を与えるため、大規模な強化学習はソフトウェアエンジニアリングタスクに広く適用されていません。その結果、DeepSeek-R1はソフトウェアエンジニアリングベンチマークでDeepSeek-V3に対して大きな改善を示していません。今後のバージョンでは、ソフトウェアエンジニアリングデータでの棄却サンプリングの実装または強化学習プロセス中の非同期評価の組み込みにより、これに対応する予定です。

参考文献

AI@Meta. Llama 3.1 model card、2024。URL https://github.com/meta-llama/llama-models/blob/main/models/llama3_1/MODEL_CARD.md 。

Anthropic. Claude 3.5 sonnet、2024。URL https://www.anthropic.com/news/claude-3-5-sonnet 。

M. Chen、J. Tworek、H. Jun、Q. Yuan、H. P. de Oliveira Pinto、J. Kaplan、H. Edwards、Y. Burda、N. Joseph、G. Brockman、A. Ray、R. Puri、G. Krueger、M. Petrov、H. Khlaaf、G. Sastry、
P. Mishkin、B. Chan、S. Gray、N. Ryder、M. Pavlov、A. Power、L. Kaiser、M. Bavarian、C. Winter、P. Tillet、F. P. Such、D. Cummings、M. Plappert、F. Chantzis、E. Barnes、E. Herbert-Voss、W. H. Guss、A. Nichol、A. Paino、N. Tezak、J. Tang、I. Babuschkin、S. Balaji、S. Jain、W. Saunders、C. Hesse、A. N. Carr、J. Leike、J. Achiam、V. Misra、E. Morikawa、A. Radford、M. Knight、M. Brundage、M. Murati、K. Mayer、P. Welinder、B. McGrew、D. Amodei、
S. McCandlish、I. Sutskever、W. Zaremba。コードでトレーニングされた大規模言語モデルの評価。CoRR、abs/2107.03374、2021。URL https://arxiv.org/abs/2107.03374 。

A. Dubey、A. Jauhri、A. Pandey、A. Kadian、A. Al-Dahle、A. Letman、A. Mathur、A. Schelten、A. Yang、A. Fan、他。llama 3群のモデル。arXiv preprint arXiv:2407.21783、2024。

Y. Dubois、B. Galambosi、P. Liang、T. B. Hashimoto。長さ制御AlpacaEval:自動評価器のバイアスを除去する簡単な方法。arXiv preprint arXiv:2404.04475、2024。

X. Feng、Z. Wan、M. Wen、S. M. McAleer、Y. Wen、W. Zhang、J. Wang。AlphaZeroのような木探索は大規模言語モデルのデコーディングとトレーニングを誘導できます、2024。URL https://arxiv.org/abs/2309.17179 。

L. Gao、J. Schulman、J. Hilton。報酬モデルの過最適化のスケーリング則、2022。URL https://arxiv.org/abs/2210.10760 。

A. P. Gema、J. O. J. Leang、G. Hong、A. Devoto、A. C. M. Mancino、R. Saxena、X. He、Y. Zhao、X. Du、M. R. G. Madani、C. Barale、R. McHardy、J. Harris、J. Kaddour、E. van Krieken、
P. Minervini。私たちはMMLUで完成したのでしょうか?CoRR、abs/2406.04127、2024。URL https://doi.org/10.48550/arXiv.2406.04127 。

Google。次世代モデル:Gemini 1.5、2024。URL https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024 。

Y. He、S. Li、J. Liu、Y. Tan、W. Wang、H. Huang、X. Bu、H. Guo、C. Hu、B. Zheng、他。中国のSimpleQA:大規模言語モデルに対する中国語の事実性評価。arXiv preprint arXiv:2411.07140、
2024。

D. Hendrycks、C. Burns、S. Basart、A. Zou、M. Mazeika、D. Song、J. Steinhardt。膨大なマルチタスク言語理解の測定。arXiv preprint arXiv:2009.03300、2020。

Y. Huang、Y. Bai、Z. Zhu、J. Zhang、J. Zhang、T. Su、J. Liu、C. Lv、Y. Zhang、J. Lei、他。C-Eval:財団モデル向けのマルチレベルマルチ規律の中国語評価スイート。arXiv preprint arXiv:2305.08322、2023。

N. Jain、K. Han、A. Gu、W. Li、F. Yan、T. Zhang、S. Wang、A. Solar-Lezama、K. Sen、I. Stoica。LiveCodeBench:大規模言語モデルのためのホリスティック汚染フリー評価。CoRR、abs/2403.07974、2024。URL https://doi.org/10.48550/arXiv.2403.07974 。

S. Krishna、K. Krishna、A. Mohananey、S. Schwarcz、A. Stambler、S. Upadhyay、M. Faruqui。事実、取得、推論:検索拡張生成の統一評価。CoRR、abs/2409.12941、2024。doi:10.48550/ARXIV.2409.12941。URL https://doi.org/10.48550/arXiv.2409.12941 。

A. Kumar、V. Zhuang、R. Agarwal、Y. Su、J. D. Co-Reyes、A. Singh、K. Baumli、S. Iqbal、C. Bishop、R. Roelofs、他。強化学習を通じて自己修正するように言語モデルをトレーニングする。arXiv preprint arXiv:2409.12917、2024。

H. Li、Y. Zhang、F. Koto、Y. Yang、H. Zhao、Y. Gong、N. Duan、T. Baldwin。CMMLU:中国語での膨大なマルチタスク言語理解の測定。arXiv preprint arXiv:2306.09212、2023。

T. Li、W.-L. Chiang、E. Frick、L. Dunlap、T. Wu、B. Zhu、J. E. Gonzalez、I. Stoica。クラウドソース データから高品質ベンチマークへ:Arena-HardおよびBenchBuilder パイプライン。arXiv preprint arXiv:2406.11939、2024。

H. Lightman、V. Kosaraju、Y. Burda、H. Edwards、B. Baker、T. Lee、J. Leike、J. Schulman、I. Sutskever、K. Cobbe。ステップバイステップで検証しましょう。arXiv preprint arXiv:2305.20050、2023。

B. Y. Lin。ZeroEval:言語モデルを評価するための統一フレームワーク、2024年7月。URL https://github.com/WildEval/ZeroEval 。

MAA。アメリカンインビテーショナル数学試験 - AIME。アメリカンインビテーショナル数学試験 - AIME 2024では、2024年2月。URL https://maa.org/math-competitions/american-invitational-mathematics-examination-aime 。

OpenAI。こんにちはGPT-4o、2024a。URL https://openai.com/index/hello-gpt-4o/ 。

OpenAI。大規模言語モデルで推論することを学ぶ、2024b。URL https://openai.com/index/learning-to-reason-with-llms/ 。

OpenAI。SimpleQAの紹介、2024c。URL https://openai.com/index/introducing-simpleqa/ 。

OpenAI。SWE-bench検証の導入、人間が検証したSWEbenchのサブセットをリリースしている、2024d。URL https://openai.com/index/introducing-swe-bench-verified/ 。

Qwen。Qwq:未知の境界を深く考える、2024a。URL https://qwenlm.github.io/blog/qwq-32b-preview/ 。

Qwen。Qwen2.5:ファウンデーションモデルのパーティー、2024b。URL https://qwenlm.github.io/blog/qwen2.5 。

D. Rein、B. L. Hou、A. C. Stickland、J. Petty、R. Y. Pang、J. Dirani、J. Michael、S. R. Bowman。GPQA:大学院レベルのGoogle証明質問応答ベンチマーク。arXiv preprint arXiv:2311.12022、2023。

Z. Shao、P. Wang、Q. Zhu、R. Xu、J. Song、M. Zhang、Y. Li、Y. Wu、D. Guo。DeepSeekMath:オープン言語モデルにおける数学的推論の限界を押し上げる。arXiv preprint arXiv:2402.03300、2024。

D. Silver、T. Hubert、J. Schrittwieser、I. Antonoglou、M. Lai、A. Guez、M. Lanctot、L. Sifre、D. Kumaran、T. Graepel、T. P. Lillicrap、K. Simonyan、D. Hassabis。自己対話による一般的な強化学習アルゴリズムでチェスと将棋をマスターする。CoRR、abs/1712.01815、2017a。URL http://arxiv.org/abs/1712.01815 。

D. Silver、J. Schrittwieser、K. Simonyan、I. Antonoglou、A. Huang、A. Guez、T. Hubert、L. Baker、M. Lai、A. Bolton、Y. Chen、T. P. Lillicrap、F. Hui、L. Sifre、G. van den Driessche、
T. Graepel、D. Hassabis。人間の知識なしでゲームを支配する。Nat.、550(7676):354-359、2017b。doi:10.1038/NATURE24270。URL https://doi.org/10.1038/nature24270 。

C. Snell、J. Lee、K. Xu、A. Kumar。LLMテスト時計算を最適にスケーリングすることは、モデルパラメータをスケーリングするよりも効果的です、2024。URL https://arxiv.org/abs/2408.03314 。

T. Trinh、Y. Wu、Q. Le、H. He、T. Luong。人間のデモンストレーションなしでオリンピック幾何学を解く。Nature、2024。doi:10.1038/s41586-023-06747-5。

J. Uesato、N. Kushman、R. Kumar、F. Song、N. Siegel、L. Wang、A. Creswell、G. Irving、I. Higgins。プロセスベースおよび結果ベースのフィードバックで数学の単語問題を解く。arXiv preprint arXiv:2211.14275、2022。

P. Wang、L. Li、Z. Shao、R. Xu、D. Dai、Y. Li、D. Chen、Y. Wu、Z. Sui。Math-Shepherd:数学的推論におけるLLMのラベルフリーステップバイステップ検証機。arXiv preprint arXiv:2312.08935、2023。

X. Wang、J. Wei、D. Schuurmans、Q. Le、E. Chi、S. Narang、A. Chowdhery、D. Zhou。自己一貫性は言語モデルの思考の連鎖推論を改善します。arXiv preprint arXiv:2203.11171、2022。

Y. Wang、X. Ma、G. Zhang、Y. Ni、A. Chandra、S. Guo、W. Ren、A. Arulraj、X. He、Z. Jiang、T. Li、M. Ku、K. Wang、A. Zhuang、R. Fan、X. Yue、W. Chen。MMLU-Pro:より堅牢でチャレンジングなマルチタスク言語理解ベンチマーク。CoRR、abs/2406.01574、2024。URL https://doi.org/10.48550/arXiv.2406.01574 。

C. S. Xia、Y. Deng、S. Dunn、L. Zhang。Agentless:LLMベースのソフトウェアエンジニアリングエージェントを解明する。arXiv preprint、2024。

H. Xin、Z. Z. Ren、J. Song、Z. Shao、W. Zhao、H. Wang、B. Liu、L. Zhang、X. Lu、Q. Du、
W. Gao、Q. Zhu、D. Yang、Z. Gou、Z. F. Wu、F. Luo、C. Ruan。DeepSeek-Prover-v1.5:証明アシスタントフィードバックを活用した強化学習およびモンテカルロ木探索。arXiv preprint arXiv:2408.08152、2024。URL https://arxiv.org/abs/2408.08152 。

J. Zhou、T. Lu、S. Mishra、S. Brahma、S. Basu、Y. Luan、D. Zhou、L. Hou。大規模言語モデルの指示追従評価。arXiv preprint arXiv:2311.07911、2023。

付録

A.貢献と謝辞

主要貢献者Daya
GuoDejian
YangHaowei
ZhangJunxiao
SongRuoyu
ZhangRunxin
XuQihao
ZhuShirong
MaPeiyi
WangXiao
BiXiaokang
ZhangXingkai
YuYu WuZ.F.
WuZhibin
GouZhihong
ShaoZhuoshu
LiZiyi Gao

貢献者Aixin
LiuBing
XueBingxuan
WangBochao
WuBei
FengChengda
LuChenggang
ZhaoChengqi
DengChong
RuanDamai
DaiDeli
ChenDongjie
JiErhang
LiFangyun
LinFucong
DaiFuli
Luo*Guangbo
HaoGuanting
ChenGuowei
LiH.
ZhangHanwei
XuHonghui
DingHuazuo
GaoHui Qu
Hui Li Jianzhong
Guo Jiashi Li
Jingchang Chen
Jingyang Yuan
Jinhao Tu Junjie
Qiu Junlong Li
J.L. Cai Jiaqi Ni
Jian Liang Jin
Chen Kai Dong
Kai Hu* Kaichao
You Kaige Gao
Kang Guan
Kexin Huang
Kuai Yu Lean
Wang Lecong
Zhang Liang
Zhao Litong
Wang Liyue
Zhang Lei Xu
Leyi Xia
Mingchuan
Zhang Minghua
Zhang Minghui
Tang Mingxu
Zhou Meng Li
Miaojun Wang
Mingming Li
Ning Tian
Panpan Huang
Peng Zhang
Qiancheng Wang
Qinyu Chen
Qiushi Du Ruiqi
Ge* Ruisong
Zhang Ruizhe
Pan Runji Wang
R.J. Chen

R.L. Jin

Ruyi Chen
Shanghao Lu
Shangyan Zhou
Shanhuang Chen
Shengfeng Ye
Shiyu Wang
Shuiping Yu
Shunfeng Zhou
Shuting Pan
S.S. Li Shuang Zhou Shaoqing
Wu Shengfeng
Ye Tao Yun
Tian Pei Tianyu Sun T. Wang
Wangding Zeng
Wen Liu
Wenfeng Liang
Wenjun Gao
Wenqin Yu*
Wentao Zhang
W.L. Xiao Wei An Xiaodong
Liu Xiaohan
Wang Xiaokang
Chen Xiaotao
Nie Xin Cheng
Xin Liu Xin Xie
Xingchao Liu
Xinyu Yang
Xinyuan Li
Xuecheng Su
Xuheng Lin
X.Q. Li
Xiangyue Jin
Xiaojin Shen
Xiaosha Chen
Xiaowen Sun
Xiaoxiang Wang Xinnan
Song Xinyi
Zhou Xianzu
Wang Xinxia
Shan Y.K. Li
Y.Q. Wang
Y.X. Wei Yang
Zhang Yanhong
Xu Yao Li Yao
Zhao Yaofeng
Sun Yaohui
Wang Yi Yu
Yichao Zhang
Yifan Shi
Yiliang Xiong
Ying He Yishi
Piao Yisong
Wang Yixuan
Tan Yiyang Ma*
Yiyuan Liu
Yongqiang Guo
Yuan Ou
Yuduan Wang
Yue Gong
Yuheng Zou
Yujia He
Yunfan Xiong
Yuxiang Luo
Yuxiang You
Yuxuan Liu
Yuyang Zhou
Y.X. Zhu
Yanping Huang
Yaohui Li Yi
Zheng Yuchen
Zhu Yunxian Ma Ying Tang
Yukun Zha
Yuting Yan Z.Z. Ren Zehui Ren
Zhangli Sha
Zhe Fu Zhean Xu Zhenda Xie
Zhengyan Zhang Zhewen
Hao Zhicheng
Ma Zhigang
Yan Zhiyu Wu
Zihui Gu

Zijia Zhu
Zijun Liu*
Zilin Li
Ziwei Xie
Ziyang Song
Zizheng Pan
Zhen Huang
Zhipeng Xu
Zhongyu Zhang
Zhen Zhang

各役割内では、著者は名前の最初の部分でアルファベット順にリストされています。*で示されている名前は、
当社のチームを離れた個人を示しています。