

Marco-o1: Towards Open Reasoning Models for Open-Ended Solutions

小林 範久 Present Square Co.,Ltd.

Marco-o1: Towards Open Reasoning Models for Open-Ended Solutions

Yu Zhao*, Huifeng Yin*, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, Kaifu Zhang

MarcoPolo Team, Alibaba International Digital Commerce

論文URL

<https://arxiv.org/abs/2411.14405>

アジェンダ

1. はじめに
2. Marco 推論データセット
3. MCTSによる解探索空間の拡張
4. 推論アクション戦略
5. 実験
6. 結論と今後の課題

1. はじめに

Marco-o1の概要

Alibaba社によって開発された大規模推論モデル（LRM：Large Reasoning Model）。OpenAI o1のように明確な基準や報酬がないオープンエンドな課題解決を目指している。

特徴

CoT（Chain-of-Thought）、**MCTS**（Monte Carlo Tree Search）、**リフレクションメカニズム**を活用し、複雑な現実課題にも対応可能。

課題

現状では、性能が OpenAI o1 モデルに及ばず、オープンエンドタスクへの完全な対応や技術的ロードマップの明確化に向けて、引き続き最適化と改良が必要である。

1. はじめに

本研究の成果

CoTデータでのファインチューニング

オープンソースのCoTデータセットと独自の合成データを組み合わせてベースモデルの全パラメータをファインチューニングし、Marco-o1-CoTを開発。

MCTSによる解探索空間の拡張

モデルの出力信頼度を利用して探索を導き、解空間を拡張するために、LLMとMCTS（Marco-o1-MCTS）を統合。

推論アクション戦略

新たな推論アクション戦略とリフレクションメカニズム（Marco-o1-MCTSミニステップ）を実装した。
MCTSフレームワーク内で異なるアクションの粒度を探索し、モデルに自己反省を促すことが含まれる。
⇒モデルの複雑な問題解決能力を大幅に向上。

翻訳タスクへの適用

機械翻訳タスクにおけるLRMを初めて調査し、多言語および翻訳分野での推論時間のスケールング則を探索。

1. はじめに

Marco-o1 の全体概要

大きく以下の3つの部分に分けられる。

- データセット
- 推論プロセス
- 信頼スコアと結論出力

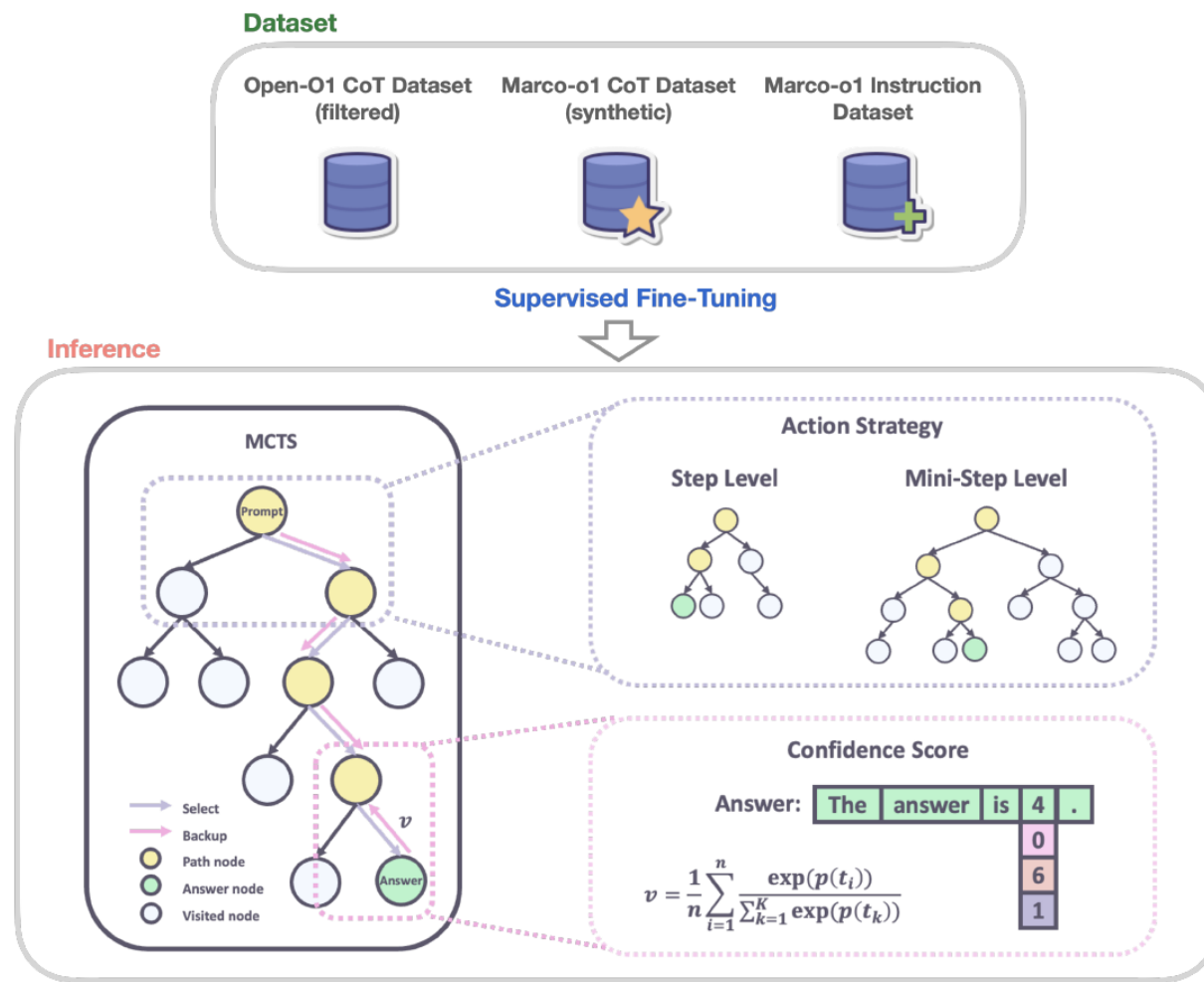


Figure 2 | The overview of Marco-o1.

1. はじめに

全体のプロセスとサンプル問題

1. プロンプト提示

1. モデルは、「strawberryという単語に含まれる 'r' の数はいくつか？」という質問を受け取る。
2. これを起点として推論ツリーを形成する。

2. タスクの分割

1. モデルはタスクを分割し、次の2つのステップに分ける。
 1. "strawberry" に含まれる 'r' の数を数える。
 2. 結果を書き出し、検証する。

3. 具体的な推論

1. モデルは文字ごとに 'r' の出現をカウントします。以下のプロセスを踏む。
 1. "strawberry" の各文字を順にチェック。
 2. 1文字目の 's' は 'r' ではない。
 3. 3文字目の 'r' を最初の出現として認識。
 4. 8文字目と9文字目の 'r' を、それぞれ2回目と3回目の出現として記録。
2. その結果、「strawberry には 'r' が3つ含まれる」と結論づける。

4. 結論の再確認と表現

1. 推論結果を確認し、「strawberryには 'r' が3つ含まれる」という結論を確認。
2. 検証に基づき、最終的な回答「3」を出力する。

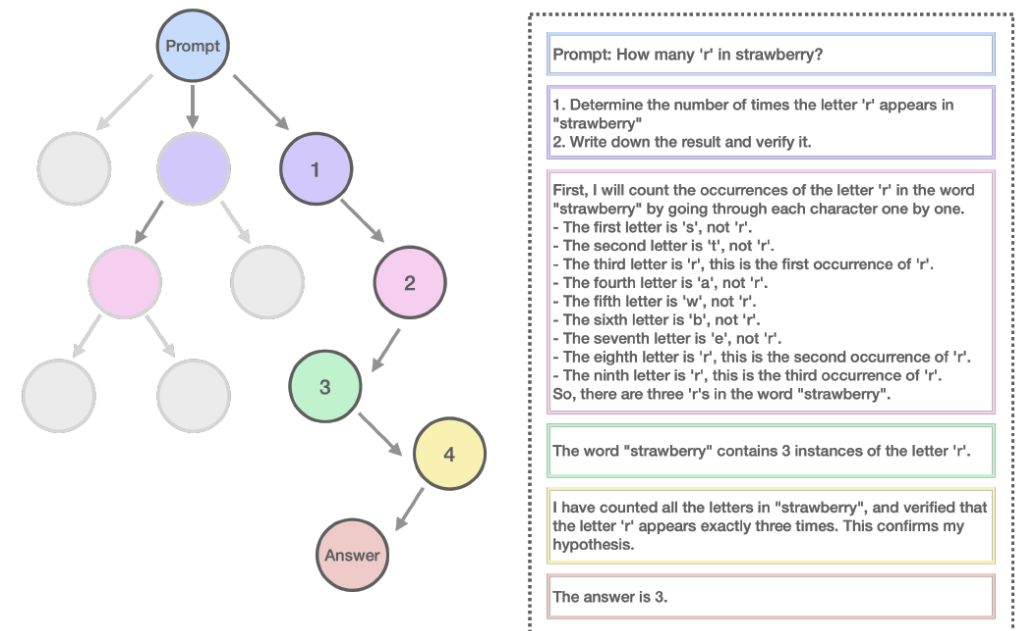


Figure 1 | A classic question reasoned by our Marco-o1 model: “How many ‘r’s are in ‘strawberry’.”

2. Marco推論データセット

Marco推論データセット

多様なデータセットを用いた教師ありファインチューニング（SFT）戦略を採用。

1. Open-O1 CoTデータセット（フィルタリング済み）

ヒューリスティックおよび品質フィルタリングプロセスを適用して、Open-O1プロジェクトのCoTデータセットを精製。モデルが構造化された推論パターンを効果的に採用できるようになる。

2. Marco-o1 CoTデータセット（合成データ）

MCTSを使用してMarco-o1 CoTデータセットを生成。複雑な推論経路の形成が可能となり、モデルの推論能力を強化する。

3. Marco Instructionデータセット

複雑なタスクを実行する際の指示追従能力の重要性を認識し、一連の指示追従データを統合。モデルのタスク遂行能力がさらに向上する。

Dataset	Number of Samples
Open-O1 CoT Dataset (Filtered) [OpenO1 Team, 2024]	45,125
Marco-o1 CoT Dataset (Synthetic)	10,000
Marco Instruction Dataset	5,141
Total	60,266

Table 1 | Overview of Marco Reasoning Datasets.

3. MCTSによる解探索空間の拡張

MCTSによる解探索空間の拡張

Marco-o1モデルは、MCTSを統合することで推論能力を強化した。この手法により、モデルはより自信があり信頼性の高い推論経路を探索できる。

1. ノードを推論状態として定義

MCTSフレームワークでは、各ノードが問題解決プロセスの推論状態を表す。

2. アクションをLLMの出力として扱う

各ノードからの可能なアクションはLLMによって生成される出力であり、これらは推論チェーン内の潜在的なステップやミニステップを示す。

3. ロールアウトと報酬計算

ロールアウトフェーズ中、LLMは推論プロセスを終端状態まで継続し、その結果をもとに報酬を計算する。

4. MCTSの誘導

報酬スコア R を用いて、有望な経路を評価し選択する。

3. MCTSによる解探索空間の拡張

信頼スコアの計算

- 各トークン t_i に対する信頼スコア c_i を以下の式で計算する。

$$c_i = \frac{\exp(p(t_i))}{\sum_{k=1}^5 \exp(p(t_k))}$$

ここで、 $p(t_i)$ はトークン t_i の対数確率、 $p(t_k)$ は上位5つの予測トークンの対数確率を表す。
このスコアは選択されたトークンが代替候補に対してどれだけ信頼性があるかを示し、スコアを0から1の範囲に正規化する。

- 次に、ロールアウトシーケンス内の全トークンの信頼スコアの平均を計算し、全体の報酬スコア v を以下の式で算出する。

$$v = \frac{1}{n} \sum_{i=1}^n c_i$$

ここで、 n はロールアウトシーケンス内のトークン数を表す。 v はロールアウト経路の品質を評価する報酬信号として機能し、値が高いほど、より信頼性が高く正確な推論経路を示す。

この手法により解探索空間が大幅に拡張され、モデルは多様な推論経路を探索し、信頼スコアに基づいて最も確率の高い経路を選択することが可能となった。

3. MCTSによる解探索空間の拡張

MCTSによる解探索空間の拡張

Marco-o1-CoTモデルと比較して、MCTSを統合した3つのモデルが改善を示している。
MCTSの導入がモデルの解探索空間を拡大し、正解を導き出す確率を向上させる効果があることが示されている。

Model	MGSM-En (Acc.)	MGSM-Zh (Acc.)
Qwen2-7B-Instruct	84.00%	76.80%
Marco-o1-CoT	85.60%	71.20%
Marco-o1-MCTS (step)	90.40%	80.00%
Marco-o1-MCTS (mini-step of 64 tokens)	88.40%	80.40%
Marco-o1-MCTS (mini-step of 32 tokens)	87.60%	82.40%

Table 2 | Experimental results on MGSM datasets.

4. 推論アクション戦略

4.1 Reasoning Action Strategyの概要

課題: MCTSの探索でアクションの粒度が粗いと、複雑な問題解決に必要な微細な推論経路を見逃す可能性がある。

解決策: 探索の粒度を調整することで、モデルの探索空間を拡大し、問題解決能力を強化する。

2つのアクション戦略

1. Step as Action (ステップ単位のアクション)

- 推論の各ステップやアクションラベルを1つの単位として扱う。
- 効率的な探索が可能だが、複雑な問題解決に必要な細かな推論経路を見逃す可能性がある。

2. Mini-step as Action (ミニステップ単位のアクション)

- ステップを64トークンまたは32トークンの小単位（ミニステップ）に分割。
- 探索の解像度が向上し、複雑な推論タスクをより効果的に処理可能になる。
- ただし、計算リソースの増加や効果的な報酬モデル設計の課題が存在する。

4. 推論アクション戦略

4.2 Reflection after Thinkingの概要

リフレクションメカニズム

- 各推論プロセスの最後に「Wait! Maybe I made some mistakes! I need to rethink from scratch.」というフレーズを追加することで、モデル自身に自己反省と推論ステップの再評価を促す仕組みを導入。

効果と改善点

1. 精度の向上

- 元々誤って解答していた困難な問題に対して、大幅な改善をもたらした。
- リフレクションを導入することで、対象の問題の約半数が正しく解答されるようになった。

2. 自己批判的視点の活用

- モデルが自らの推論に潜在的なエラーを見つけ出す能力を高める。
- 初期結論を問い直し、推論プロセスを再表現・精緻化することを奨励する。

3. 信頼性の向上

- 矛盾やミスを検出する能力を活用することで、より正確で信頼性の高い問題解決が可能となった。

5. 実験

5.1 実験設定の概要

1. モデルのベース

- Qwen2-7B-Instructを基に、トレーニングデータを使用してMarco-o1-CoTを作成。
- Marco-o1-CoTをMCTSフレームワーク内で実験的に使用。

2. MCTSの3つの構成

- Marco-o1-MCTS (mini-step of 64 tokens): 64トークンのミニステップをアクションとして扱う。
- Marco-o1-MCTS (mini-step of 32 tokens): 32トークンのミニステップをアクションとして扱う。

3. テスト環境

- テストには英語 (En) と中国語 (Zh) のMGSMデータセットを使用。
- 各モデルは、CoTプロンプトを活用して推論プロセスの一貫性を確保。
- Marco-o1-MCTS (step): 推論の各ステップを1つのアクションとして扱う。

5. 実験

5.2 結果

1. プロンプト提示
2. タスクの分割
3. 具体的な推論
4. 結論の再確認と表現

結果

- 「strawberry」という単語に含まれる「r」の数を問う質問に対し、モデルは正しい答えを出した。
- モデルは推論過程で最後の文字「y」に言及していなかった。
- この省略はタイプミスや意図的な見落としではないと確認された。

考察

- 多くの検索ルートが「y」を無視する一方、一部のルートでは「y」を考慮していた。
- この現象は、モデルが人間のように自信を持ちすぎる傾向を持ち、明白なステップを省略する可能性を示唆している。
⇒ 異なる言語や設定におけるモデルの推論能力強化の可能性を示している。
(重要な部分に集中し、他の部分を省略する柔軟性を持っている可能性がある。)

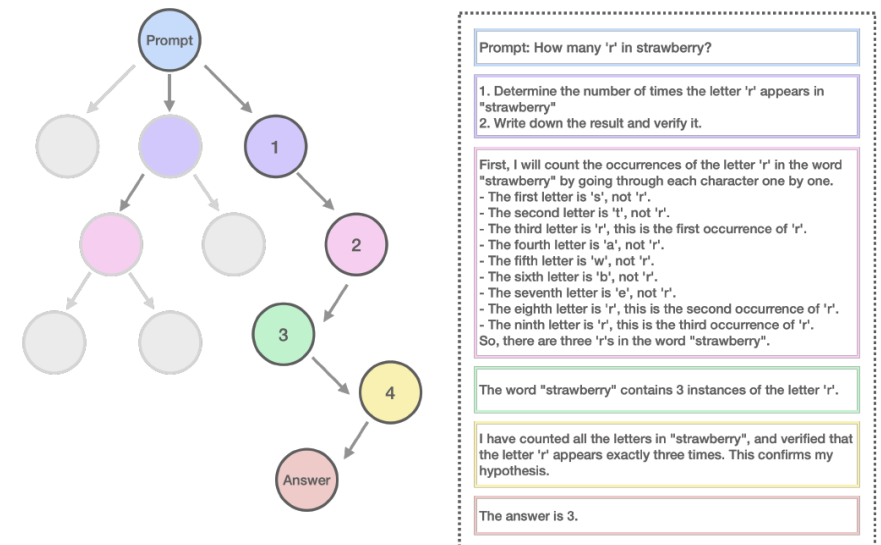


Figure 1 | A classic question reasoned by our Marco-o1 model: "How many 'r's are in 'strawberry'."

5. 実験

5.2 結果

- Chain-of-Thought(CoT)では、部分的に正しい推論が可能だが、誤った仮定を含むため最終的な結論が間違っている。
- モンテカルロ木探索 (MCTS) を組み合わせたことで、問題の構造を段階的に再評価し、全体として正しい結論を導けるようになった。
- MCTSの導入により解答探索空間が広がり、正しい解答に到達する可能性が大幅に向上した。

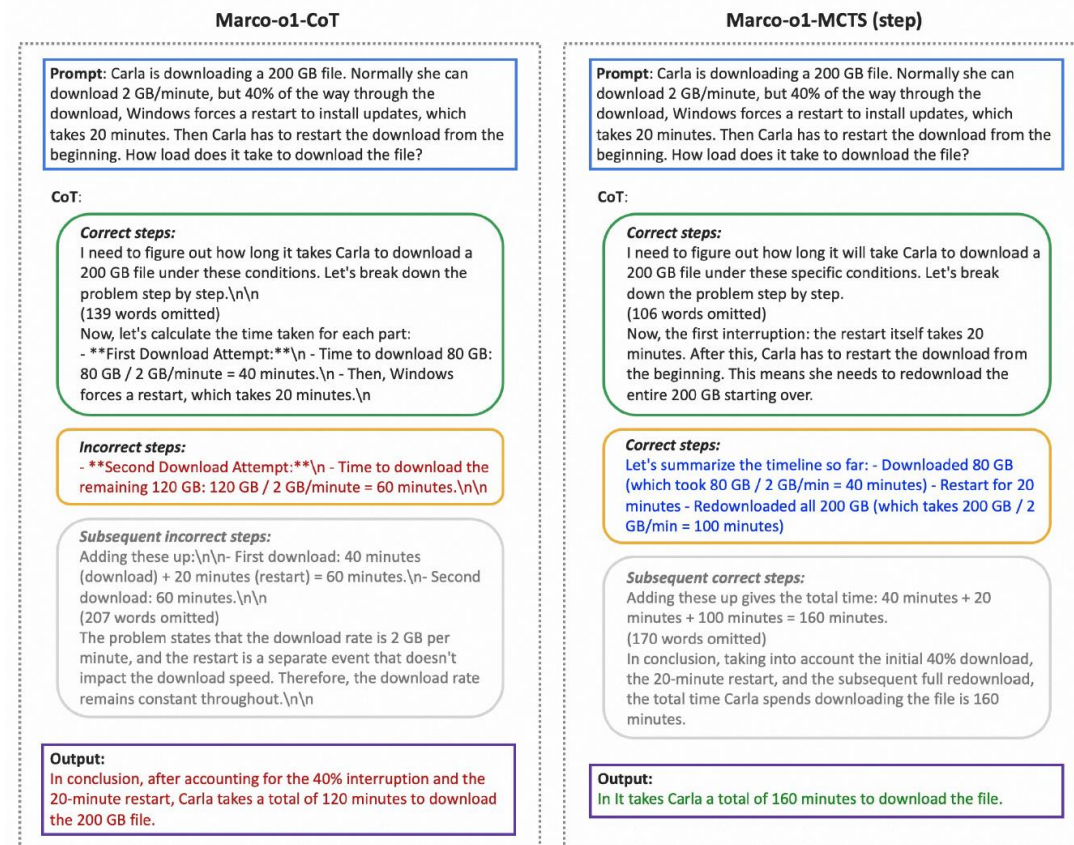


Figure 4 | MCTS Expands the Solution Space for Correct Answers. Comparison between Marco-o1-CoT (left) and Marco-o1-MCTS (step) (right) on the MGSM dataset. While Marco-o1-CoT fails to provide the correct answer, integrating MCTS with step-level actions allows the model to explore a broader solution space, increasing the likelihood of arriving at the correct solution.

5. 実験

5.2 結果

プロンプト

Carlaは200GBのファイルをダウンロードしています。通常、2GB/分の速度でダウンロードできますが、ダウンロードの40%が完了したところで、Windowsがアップデートをインストールするために再起動を強制します。この再起動には20分かかります。その後、ダウンロードを最初からやり直す必要があります。このファイルのダウンロードにはどれだけ時間がかかりますか？

正しい手順

• Carlaがこの特定の条件下で200GBのファイルをダウンロードするのにどれくらい時間がかかるかを確認する必要がある。問題をステップごとに分解する。(以下、139語が省略されています)

• 最初のダウンロード試行

- 80GBをダウンロードするのにかかる時間は以下の通り
 $80\text{GB} \div 2\text{GB/分} = 40\text{分}$
- その後、Windowsが再起動を強制し、再起動には**20分**かかる。

誤った手順

• 2回目のダウンロード試行

- 残りの120GBをダウンロードするのにかかる時間は以下の通り
 $120\text{GB} \div 2\text{GB/分} = 60\text{分}$

出力

「結論として、40%の中断と再起動の20分を考慮すると、Carlaが200GBのファイルをダウンロードするのにかかる時間は合計で**120分**です。」

✗ 問題文では、再起動後にダウンロードが「最初からやり直し」になると述べられているが、モデルはこれを考慮せず「途中から再開する」と仮定している。

Marco-o1-CoT

Prompt: Carla is downloading a 200 GB file. Normally she can download 2 GB/minute, but 40% of the way through the download, Windows forces a restart to install updates, which takes 20 minutes. Then Carla has to restart the download from the beginning. How long does it take to download the file?

CoT:

Correct steps:

I need to figure out how long it takes Carla to download a 200 GB file under these conditions. Let's break down the problem step by step.\n\n(139 words omitted)

Now, let's calculate the time taken for each part:

- **First Download Attempt:**\n - Time to download 80 GB: $80\text{ GB} / 2\text{ GB/minute} = 40\text{ minutes}$.
 - Then, Windows forces a restart, which takes 20 minutes.\n

Incorrect steps:

- **Second Download Attempt:**\n - Time to download the remaining 120 GB: $120\text{ GB} / 2\text{ GB/minute} = 60\text{ minutes}$.
 -

Subsequent incorrect steps:

Adding these up:\n\n - First download: 40 minutes (download) + 20 minutes (restart) = 60 minutes.
 - Second download: 60 minutes.
 - (207 words omitted)

The problem states that the download rate is 2 GB per minute, and the restart is a separate event that doesn't impact the download speed. Therefore, the download rate remains constant throughout.\n\n

Output:

In conclusion, after accounting for the 40% interruption and the 20-minute restart, Carla takes a total of 120 minutes to download the 200 GB file.

5. 実験

5.2 結果

プロンプト

Carlaは200GBのファイルをダウンロードしています。通常は2GB/分の速度でダウンロードできますが、ダウンロードの途中（40%完了時）でWindowsがアップデートをインストールするために再起動を要求します。この再起動には20分かかり、ダウンロードは最初からやり直しになります。このファイルのダウンロードにはどれくらい時間がかかるか？

正しいステップ

Carlaがこの条件下で200GBのファイルをダウンロードするのにどれくらい時間がかかるかを計算する必要がある。問題をステップごとに分解する。（106語が省略されています）最初の中断：再起動自体に20分かかる。その後、ダウンロードは最初からやり直しとなるため、再度200GB全体をダウンロードする必要がある。

(次の) 正しいステップ

これらを合計すると、

- 最初のダウンロード: 40分
- 再起動: 20分
- 再ダウンロード: 100分
- 合計時間: 160分

出力

40%のダウンロード、再起動の20分、そして再ダウンロードを考慮すると、Carlaがファイルをダウンロードするのにかかる時間は160分となる。

Marco-o1-MCTS (step)

Prompt: Carla is downloading a 200 GB file. Normally she can download 2 GB/minute, but 40% of the way through the download, Windows forces a restart to install updates, which takes 20 minutes. Then Carla has to restart the download from the beginning. How long does it take to download the file?

CoT:

Correct steps:

I need to figure out how long it will take Carla to download a 200 GB file under these specific conditions. Let's break down the problem step by step.
(106 words omitted)

Now, the first interruption: the restart itself takes 20 minutes. After this, Carla has to restart the download from the beginning. This means she needs to redownload the entire 200 GB starting over.

Correct steps:

Let's summarize the timeline so far: - Downloaded 80 GB (which took 80 GB / 2 GB/min = 40 minutes) - Restart for 20 minutes - Redownloaded all 200 GB (which takes 200 GB / 2 GB/min = 100 minutes)

Subsequent correct steps:

Adding these up gives the total time: 40 minutes + 20 minutes + 100 minutes = 160 minutes.
(170 words omitted)

In conclusion, taking into account the initial 40% download, the 20-minute restart, and the subsequent full redownload, the total time Carla spends downloading the file is 160 minutes.

Output:

In It takes Carla a total of 160 minutes to download the file.

5. 実験

5.2 結果

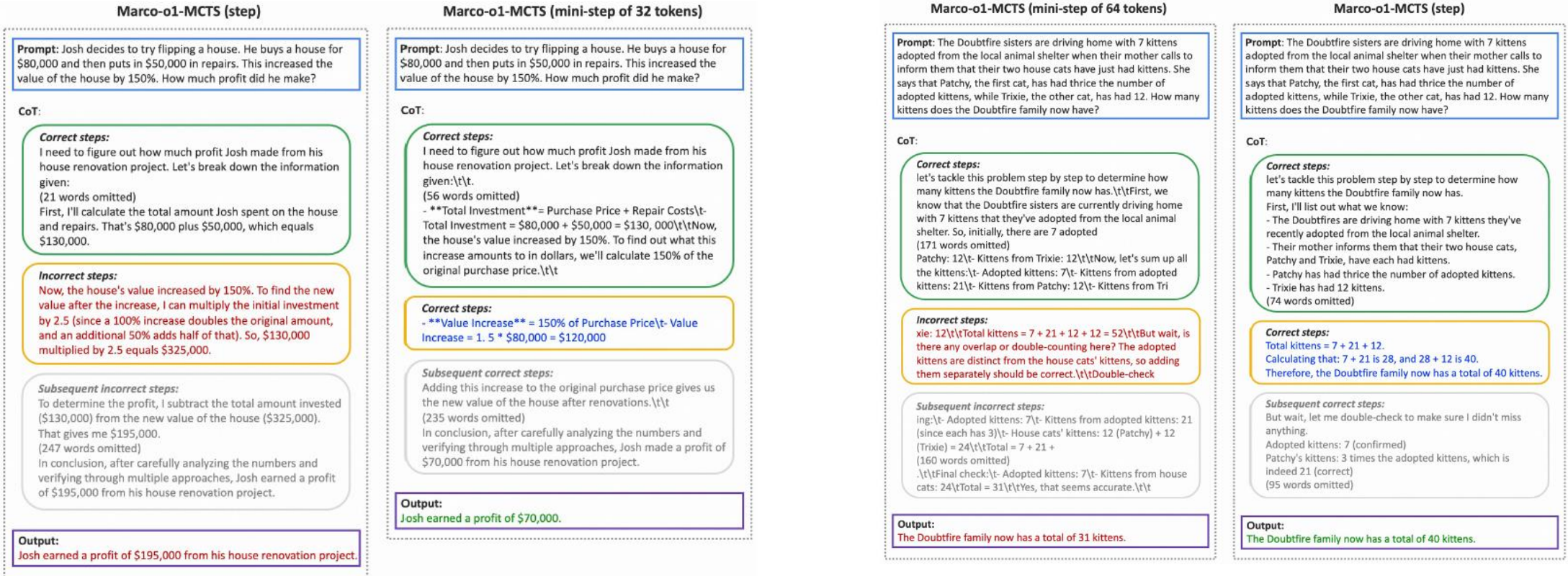


Figure 5 | Finer Granularity with mini-steps Enhances Problem-Solving. Comparison between Marco-o1-MCTS (step) (left) and Marco-o1-MCTS (mini-step of 32 tokens) (right) on the MGSM dataset. The step-level action strategy does not yield the correct answer, but by using a finer-grained mini-step of 32 tokens, the model successfully navigates the solution space to find the correct answer, demonstrating the effectiveness of increased action granularity.

Figure 6 | Optimal Action Granularity Depends on Problem Complexity. Comparison between Marco-o1-MCTS (mini-step of 64 tokens) (left) and Marco-o1-MCTS (step) (right) on the MGSM dataset. The model with a mini-step of 64 tokens fails to find the correct answer, whereas using step-level actions enabled the model to correctly solve the problem. This highlights that we cannot draw definitive conclusions about which action strategy is superior. We believe that as the reward becomes more accurate, the larger solution space provided by MCTS will demonstrate greater potential.

5. 実験

5.3 翻訳タスクに関するケーススタディ

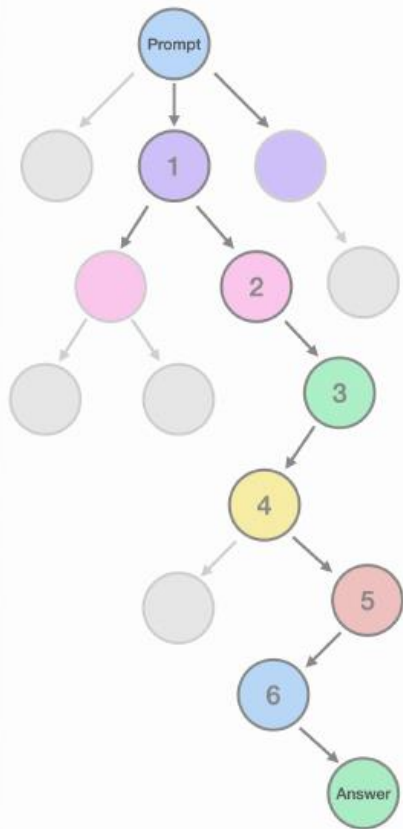
Marco-o1モデルが翻訳タスクにおいてどのような能力を持つかを実証するため、Google Translateとの比較ケーススタディを実施。特に、口語的表現やスラングを翻訳する際の性能を評価。

結果

- Marco-o1は複雑なスラングや口語的表現を処理する能力において、Google Translateを上回る性能を示した。
- 文脈の理解、ニュアンスの把握、自然な翻訳結果の生成という点でMarco-o1の優れた性能が確認された。

5. 実験

5.3 翻訳タスクに関するケーススタディ



Prompt: 翻译到英文: 这个鞋拥有踩屎感, 很舒服, 推荐购买

Okay, I need to translate the sentence "这个鞋拥有踩屎感, 很舒服, 推荐购买." into English.

First, I'll break down the sentence to understand each part.
- "这个鞋" means "this shoe," so that's straightforward.
- "拥有踩屎感" is a bit tricky. "拥有" means "to have" or "to possess," and "踩屎感" is a colloquial term that roughly translates to "feeling of stepping on feces." That's not the most polite way to describe a shoe, so I need to find a more appropriate English equivalent. Maybe "comfortable feeling" or "soft sole" would be better.
- "很舒服" directly translates to "very comfortable," which is a positive attribute.
- "推荐购买" means "recommend purchasing."

Putting it all together, the sentence is saying that the shoe is comfortable and recommends buying it, but the first part about "踩屎感" is a bit crude.

Perhaps I should adjust that part to make it more suitable for an English-speaking audience, maybe something like "the shoe offers a comfortable sole" instead of "has a feeling of stepping on feces." that way, it maintains the positive aspect while avoiding the crude language.

So, the translated sentence would be: "This shoe offers a comfortable sole and is highly recommended for purchase."

But to make it sound more natural in English, I could rephrase it as: "This shoe has a comfortable sole and is highly recommended for purchase." that sounds more fluent and avoids any offensive terms.

So, the final translation would be: This shoe has a comfortable sole and is highly recommended for purchase.

Figure 7 | Demonstration of translation task using Marco-o1 of a colloquial expression "This shoe has a comfortable sole and is highly recommended for purchase".

図の構造と翻訳プロセス

1. プロンプト

1. 入力フレーズ: 「这个鞋拥有屎感很服, 推荐购买」
2. モデルはこれを英語に翻訳するというタスクを受け取る。

2. 推論のステップ分解

Step 1: フレーズを部分ごとに分解し、意味を理解する。

1. 「这个鞋」→「This shoe」
2. 「拥有屎感」→「feeling of stepping on feces」(直訳だが適切ではない)
3. 「很服」→「very comfortable」
4. 「推荐购买」→「recommend purchasing」

Step 2: 適切な英訳を選択する。

1. 「拥有屎感」を「comfortable feeling」や「soft sole」に置き換えるなど、英語圏の文化や適切な表現に基づいて再構成する。

Step 3: 全体の意味をまとめる。

1. フレーズ全体が「靴が快適で購入を勧める」という内容を表しているを理解する。

Step 4: より自然な英語表現に調整する。

1. 「The shoe offers a comfortable sole」という表現を採用し、「粗野な言葉」を避ける。

Step 5: より流暢で洗練された最終的な翻訳を生成する。

1. **最終翻訳:** 「This shoe has a comfortable sole and is highly recommended for purchase.」

5. 実験

5.3 翻訳タスクに関するケーススタディ

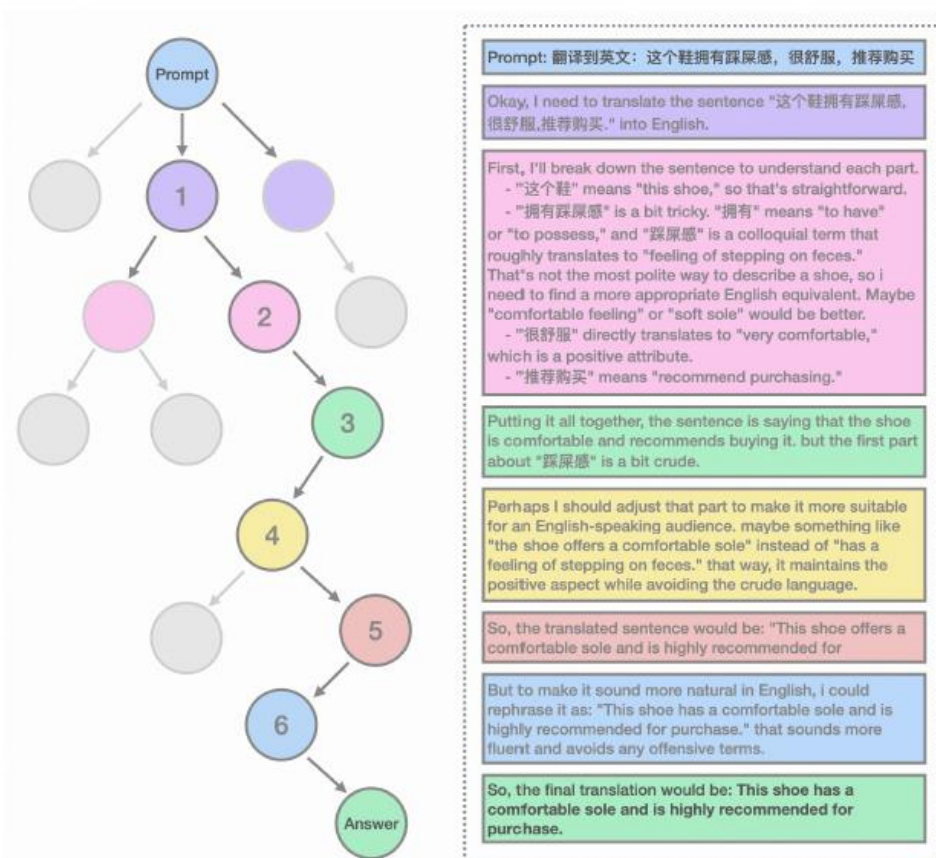


Figure 7 | Demonstration of translation task using Marco-o1 of a colloquial expression “This shoe has a comfortable sole and is highly recommended for purchase”.

ポイント

1. 口語的な表現の処理

Marco-o1は、文化的に適切でない表現（例：「屎感」）を検出し、英語の文脈に合ったポジティブな表現（例：「comfortable sole」）に置き換える。

2. 段階的な推論

フレーズの意味理解から、部分的な翻訳、全体の再構成、自然な表現の選択まで、段階的に処理を進める。

3. 最終翻訳の精度

単なる直訳ではなく、文化的・文脈的に適切で流暢な英訳を生成している。

5. 実験

5.3 翻訳タスクに関するケーススタディ

- Marco-o1-MCTSモデルが、従来のMarco-o1-CoTやQwen2-7B-Instructを上回る精度を達成している。粒度（Granularity）の異なるアクション戦略（Step、Mini-step）が結果に影響を与えている。

1. 言語による最適な戦略の違い

英語では「Step as Action」が、中国語では「Mini-step as Action (32)」が最適な戦略である。言語特有の構造や特性に応じた最適な粒度が異なることが示唆される。

2. Marco-o1-MCTSの優位性

全体として、MCTSの統合がCoTや従来のモデルに対して大幅な精度向上をもたらしている。

3. 課題と展望

粒度の選択がモデルの性能に大きく影響を与えるため、データセットやタスクに応じた最適な戦略のさらなる研究が必要である。

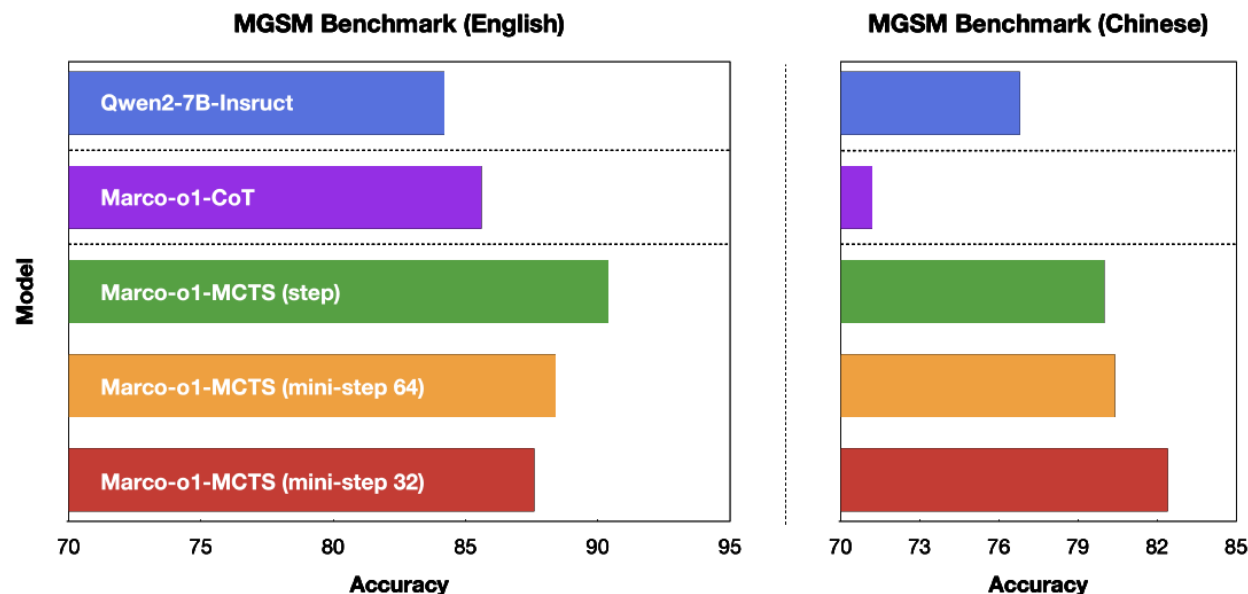


Figure 3 | The main results of Marco-o1.

5. 実験

5.3 翻訳タスクに関するケーススタディ

モデルの解答精度や探索能力を多様な条件で比較するため、Test@N（各問題に対してモデルが最大でN回の予測を行うことが許される場合、正解を1回以上出力する割合を指す。）にて比較。

MGSM-En（英語データセット）

1. Test@1

1. **Marco-o1-MCTS (step)**が最高精度（90.40%）を記録している。
2. Mini-step（64トークンまたは32トークン）よりも「Step as Action」が効果的である。

2. Test@8およびTest@32

1. すべてのMCTSモデルが高い精度（約99%以上）を示している。
2. 特にTest@8以降では、Marco-o1-CoTとの差が縮まり、ほぼ同等の結果を示す。

MGSM-Zh（中国語データセット）

1. Test@1

1. **Marco-o1-MCTS (mini-step of 32 tokens)**が最も高い精度（82.40%）を示している。
2. 英語と異なり、ミニステップの細かい粒度が中国語において有効であることが示唆される。

2. Test@8およびTest@32

1. Test@32では、すべてのモデルが高い精度（92%以上）を達成しているが、MCTSの導入が一貫して精度向上に寄与している。

Test@1ではモデル間の差が明確であり、MCTSの影響が顕著である。一方、Test@8以降ではすべてのモデルが高い精度を達成し、差が縮小する。

Model	MGSM-En		
	Test@1 Acc.	Test@8 Acc.	Test@32 Acc.
Qwen2-7B-Instruct	84.00%	89.60%	96.00%
Marco-o1-CoT	85.60%	97.60%	99.20%
Marco-o1-MCTS (step)	90.40%	99.20%	99.20%
Marco-o1-MCTS (mini-step of 64 tokens)	88.40%	98.40%	99.60%
Marco-o1-MCTS (mini-step of 32 tokens)	87.60%	98.80%	99.20%

Model	MGSM-Zh		
	Test@1 Acc.	Test@8 Acc.	Test@32 Acc.
Qwen2-7B-Instruct	76.80%	80.80%	92.40%
Marco-o1-CoT	71.20%	93.60%	96.40%
Marco-o1-MCTS (step)	80.00%	93.60%	96.00%
Marco-o1-MCTS (mini-step of 64 tokens)	80.40%	92.80%	95.20%
Marco-o1-MCTS (mini-step of 32 tokens)	82.40%	93.20%	96.80%

Table 3 | Performance on MGSM Datasets: Test@1, Test@8, and Test@32 Results. Test@N denotes the percentage of problems solved correctly at least once when the model is allowed to make N separate guesses for each problem.

6. 結論と今後の課題

まとめ

- Marco-o1は、**Chain-of-Thought (CoT) ファインチューニング**、**モンテカルロ木探索 (MCTS)**、および新しい**推論アクション戦略**を統合することで、推論能力を強化した。
- MCTSの統合により解探索空間が拡大され、ステップおよびミニステップといった異なるアクション粒度を利用した実験が、より細かい探索解像度が精度向上に寄与する可能性を示した。このアプローチは推論タスクにおける大幅な改善を示し、複雑なスラング表現の翻訳においても成功を収めた。
- 今後の展望として、**Outcome Reward Modeling (ORM)** と **Process Reward Modeling (PRM)** を通じて、MCTSの報酬信号を精緻化し、ランダム性を軽減することで性能をさらに向上させることを目指す。さらに、強化学習技術を活用してMarco-o1の意思決定プロセスを微調整し、最終的には複雑な現実世界のタスクに対応する能力を向上させる。

参考文献

- OpenAI. Learning to reason with llms. <https://openai.com/index/learning-to-reason-with-llms/>, 2024. [Accessed 19-09-2024].
- K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>.
- J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- J. Huang, S. S. Gu, L. Hou, Y. Wu, X. Wang, H. Yu, and J. Han. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*, 2022.
- A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.
- K. Valmeekam, M. Marquez, and S. Kambhampati. Can large language models really improve by self-critiquing their own plans? *arXiv preprint arXiv:2310.08118*, 2023.
- T. Zhong, Z. Liu, Y. Pan, Y. Zhang, Y. Zhou, S. Liang, Z. Wu, Y. Lyu, P. Shu, X. Yu, et al. Evaluation of openai o1: Opportunities and challenges of agi. *arXiv preprint arXiv:2409.18486*, 2024.