

000 HIDDEN PATTERNS IN CHAIN-OF-THOUGHT REASON- 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053

HIDDEN PATTERNS IN CHAIN-OF-THOUGHT REASON- ING

Anonymous authors

Paper under double-blind review

ABSTRACT

Chain-of-thought (CoT) prompting is a de-facto standard technique to elicit reasoning-like responses from large language models (LLMs), allowing them to spell out individual steps before giving a final answer. While the resemblance to human-like reasoning is undeniable, the driving forces underpinning the success of CoT reasoning still remain largely unclear. In this work, we perform an in-depth analysis of CoT traces originating from competition-level mathematics questions, with the aim of better understanding how, and which parts of CoT actually contribute to the final answer. To this end, we introduce the notion of a *potential*, quantifying how much a given part of CoT increases the likelihood of a correct completion. Upon examination of reasoning traces through the lens of the potential, we identify surprising patterns including (1) its often strong non-monotonicity (due to reasoning *tangents*), (2) very sharp but sometimes tough to interpret spikes (reasoning *insights* and *jumps*) as well as (3) at times *lucky guesses*, where the model arrives at the correct answer without providing any relevant justifications before. While some of the behaviours of the potential are readily interpretable and align with human intuition (such as insights and tangents), others remain difficult to understand from a human perspective. To further quantify the reliance of LLMs on reasoning *insights*, we investigate the notion of CoT *transferability*, where we measure the potential of a weaker model under the partial CoT from another, stronger model. Indeed aligning with our previous results, we find that as little as 20% of partial CoT can “unlock” the performance of the weaker model on problems that were previously unsolvable for it, highlighting that a large part of the mechanics underpinning CoT are transferable.

1 INTRODUCTION

Chain-of-thought (CoT) reasoning (Wei et al., 2023) has led to several breakthroughs in domains ranging from mathematics (Cobbe et al., 2021; Gao et al., 2023; Luo et al., 2025; DeepSeek-AI et al., 2025) to coding (Chen et al., 2021; Austin et al., 2021; Li et al., 2022; Lozhkov et al., 2024; Rozière et al., 2024), enabling modern language models to now win gold medals at mathematical olympiads (Luong et al., 2025). The underlying idea of CoT is very simple and intuitive: let the model reason through the given problem and explain its steps before giving a final answer. This approach offers two main advantages: (1) Generating additional tokens means more computation available to the model, allowing it to implement more complex routines. (2) CoT enables the model to decompose complex problems into more manageable sub-tasks, akin to human reasoning.

The success of chain-of-thought reasoning is undeniable, yet the precise mechanisms driving it remain poorly understood. A very tempting explanation, due to their (by design) strong resemblance to human reasoning, is that LLMs similarly benefit from spelling out bigger computations more slowly, using techniques such as backtracking and verification to explore several avenues before finally arriving at the best answer (Zhou et al., 2023; Shinn et al., 2023; Madaan et al., 2023b; Press et al., 2023). Other works however suggest that the content of CoTs might not always reflect the actual solving strategy of the model, for instance Lanham et al. (2023); Chen et al. (2025b) show that the model’s explanations to addition task do not line up with the underlying computation performed internally. This result seems to rather suggest that CoTs primarily act as computational mechanisms, letting the model execute more complicated algorithms or heuristics “under the hood” while at the same time mimicking human reasoning.

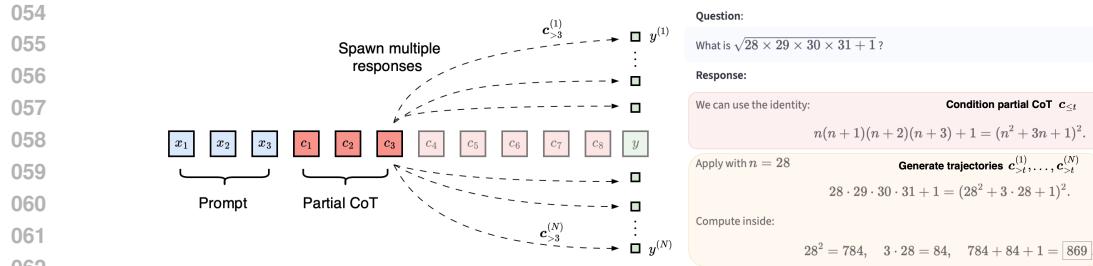


Figure 1: **Left:** Illustration of the calculation of the potential. **Right:** An example prompt and partial CoT, which in this case should intuitively raise the probability of success (i.e. the potential) significantly once discovered or provided by another model.

These perspectives motivate a closer look at how CoT actually contributes in practice. We therefore closely examine reasoning traces produced by several models with a focus on competition-level mathematics questions from AIME-2024, AIME-2025 (MAA, 2025) and MATH-500 (Hendrycks et al., 2021), GPQA-Diamond (Rein et al., 2023) for more general reasoning, as well as coding problems from HumanEval (Chen et al., 2021). In the main text we focus on AIME, as its difficult questions present an ideal arena to study properties of reasoning chains, especially as modern models still produce highly variable CoTs for the same question, *sometimes* leading to the correct solution, but often failing to do so. In this paper, we aim to understand what, or which parts of a CoT make it successful or wrong? To gain insights into this question, we introduce the notion of the *potential*, defined as the probability of success of the model when sampling conditioned on a given partial chain of the CoT (see Eq. 1 for a precise definition). As the potential initially starts out low (models can only sometimes arrive at the right answer), we can use it to monitor precisely which tokens (or collection thereof) increase or decrease it, equipping us with a tool to understand what parts of CoT unlock a previously difficult problem. We observe that similarly to humans, LLMs often exhibit reasoning *insights*, i.e., strong increases of the potential due to the completion of a conceptually difficult step (see e.g., Fig. 1, 5, 9, 11, 12, 13, 14 or 15). Not all spikes in the potential are easily interpretable however; we find that performance can significantly increase through seemingly trivial steps, coined reasoning *jumps* (see e.g. Fig. 5 or Fig. 6) Surprisingly, we observe that the potential is far from monotonic, i.e. not every token contributes effectively towards the final answer but rather long durations of no progress or even sharp drops can occur. The latter are often due to reasoning *tangents*, i.e. approaches which initially look promising but ultimately lead to dead ends or even wrong answers, (see e.g. Fig. 9, 12, 13, 14 or 15).

To further study the usage of reasoning insights in language models, we investigate the degree of *transferability* of CoT between different models. We focus on providing a weaker model with the (partial) CoT from a stronger one, with the motivation that if models indeed struggle with conceptual understanding of the problem, their reasoning might be unblocked when being provided correct sub-steps. Indeed, difficult mathematical questions often involve solving several steps of non-uniform difficulty, with some problems even becoming mostly trivial for humans once a specific insight is obtained or provided. An illustrative example of such a question is shown in Fig. 1, taken from AIME-1989 (MAA, 2025). While the question might look intimidating to many math students at first sight, the problem becomes easily solvable when presented with the insight that $n(n+1)(n+2)(n+3) + 1 = (n^2 + 3n + 1)^2$. In other words, human reasoning is often able to transfer if the gap is not too large. For LLMs, we find similar results; problems that were previously unsolved by the weaker model, gradually become solvable as more and more CoT is provided, even as little as 20% of CoT leads to a significant improvement in performance. We observe such transferability even between very different model classes, e.g. Qwen3-0.6B's accuracy significantly improves when provided with partial CoT of GPT-OSS-20B. This suggests that language models can profit from insights provided by stronger models, suggesting that some CoTs work in a model-agnostic way.

2 RELATED WORK

Chain-of-Thought reasoning has been very influential in recent years, with every modern language model now being trained to give reasoning-like responses. This characteristic has been strongly

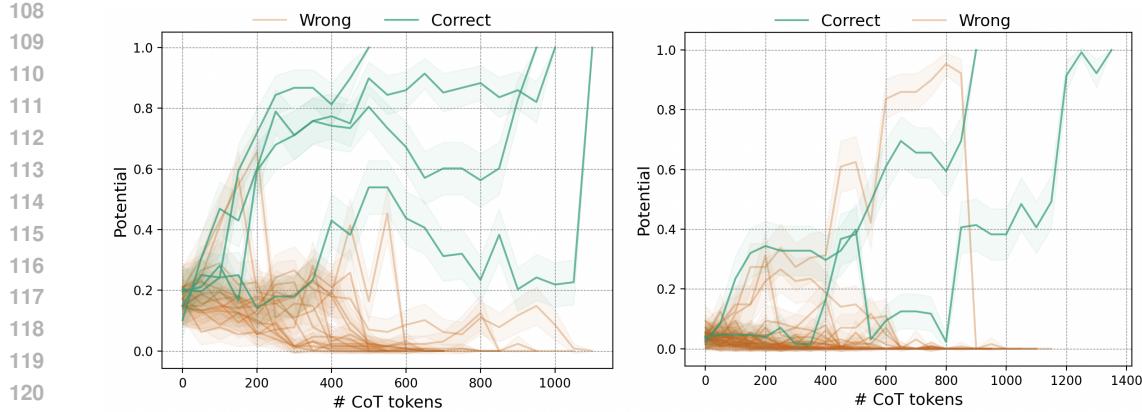


Figure 2: **Potential curves.** Potential of correct and wrong CoTs for Qwen2.5-7B on AIME-2024, Question 5 and 11. Strongly non-monotonic behaviour for both correct and incorrect CoTs.

exacerbated by the emergence of reasoning models such as o1 (OpenAI et al., 2024) and R1 (DeepSeek-AI et al., 2025), further encouraging longer responses by training with reinforcement learning with verifiable rewards. Such models now regularly require generating 128k tokens for difficult mathematics questions before returning a final answer. The still human-like nature of these reasoning chains has inspired a surge of works with the aim of interpreting and understanding how these long sequences of tokens actually contribute to the final answer. A line of work has investigated how models react when their CoT is manipulated through insertion of mistakes (Wang et al., 2023) or changes in symbols (Madaan et al., 2023a; Madaan & Yazdanbakhsh, 2022), finding them to be surprisingly robust. Other works have investigated several attribution strategies to identify important parts in CoT (Golovneva et al., 2023; Berchansky et al., 2024; Wu et al., 2023). Opposite types of findings have also been made; Lyu et al. (2023); Lanham et al. (2023); Madsen et al. (2024) have observed that CoT does not always reflect the underlying computation of the model, making it thus difficult to pin-point helpful steps in the first place. Other works go a step further and argue that CoT reasoning should not be compared to human reasoning (Kambhampati et al., 2025; Stechly et al., 2025; Bhambri et al., 2025) or that they outright imitate reasoning without actually performing any (Shojaee et al., 2025). Finally, the line of works most similar to ours also studies conditional generation from partial CoTs; Bigelow et al. (2025) investigate so-called “fork tokens” in the context of neural text generation. Bogdan et al. (2025) also explore the notion of conditional generation to find “thought anchors”, parts of CoT that help the model arrive at correct answers. While their focus is on more abstract reasoning concepts such as backtracking and self-verification, we focus on task-relevant insights and also explore the failure modes of CoT reasoning. Finally, Amani et al. (2025) also explore the notion of completing partial CoTs, incorporating the idea in reinforcement learning for better reward signal. Our latter definition of the potential shares strong resemblance to the value function in actor-critique models (Konda & Tsitsiklis, 1999; Sutton & Barto, 2018), similarly measuring the quality of a given state but in a Monte-Carlo fashion.

3 POTENTIAL OF CoT

Setup. Let \mathcal{V} denote the vocabulary. Assume we have a tokenized input prompt $\mathbf{x} \in \mathcal{V}^D$ (e.g., encoding a math question) and a ground truth answer $y^* \in \mathcal{V}$ (for simplicity represented by a single token) encoding the expected response (e.g. “513”). Let LM_θ represent a language model with parameters θ , mapping a sequence of tokens \mathbf{x} to the logits of size $|\mathcal{V}|$. When answering to a prompt, models now generate $T \in \mathbb{N}$ intermediate *chain-of-thought* tokens $\mathbf{c} \in \mathcal{V}^T$ auto-regressively before arriving at a final answer y . I.e. given prompt \mathbf{x} , we generate $c_t \sim \text{LM}_\theta(\cdot | \mathbf{c}_{<t}, \mathbf{x})$ auto-regressively and only then sample the answer, $y \sim \text{LM}_\theta(\cdot | \mathbf{c}, \mathbf{x})$. Generations involving such intermediate tokens have been observed to outperform models trained (or prompted) to directly provide an answer in a variety of settings (Wei et al., 2023). We will often abuse notation slightly by letting $(y, \mathbf{c}_{\geq t}) \sim \text{LM}_\theta(\cdot | \mathbf{c}_{<t}, \mathbf{x})$ denote the (sequential) autoregressive generation, conditional on $(\mathbf{c}_{<t}, \mathbf{x})$. Typical decoding strategies in language models leverage this stochastic generation and

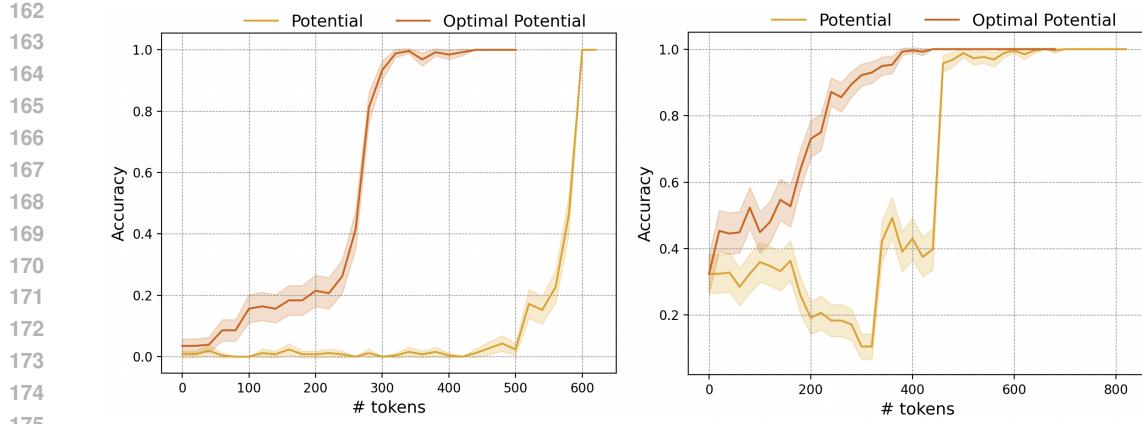


Figure 3: The potential of optimal and standard CoT for AIME-2025-I, question 1 and 5. While standard CoT eventually arrives at the right answer, optimal CoT does so in a more robust way.

it is hence interesting to consider $K \in \mathbb{N}$ such generations by varying the random seeds, either unconditionally or starting from a partial CoT $\mathbf{c}_{<t}$,

$$\left(y^{(k)}, \mathbf{c}_{\geq t}^{(k)} \right) \stackrel{i.i.d.}{\sim} \text{LM}_\theta(\cdot | \mathbf{c}_{<t}, \mathbf{x}) \quad \text{for } k = 1, \dots, K$$

where we obtain most likely distinct CoT completions $\mathbf{c}_{\geq t}^{(k)}$ and final answers $y^{(k)}$.

Potential. Given a prompt \mathbf{x} and an associated reasoning process \mathbf{c} with final answer y , it is natural to ask which sub-steps in \mathbf{c} contributed most to the overall result. Let us define the *potential* of a chunk of CoT $\mathbf{c}_{<t}$ on the prompt \mathbf{x} as the probability of correct generation conditioned on $\mathbf{c}_{<t}$,

$$\text{pot}(\mathbf{c}_{<t}; \mathbf{x}) := \mathbb{P}_{(\mathbf{c}_{\geq t}, y) \sim \text{LM}_\theta(\cdot | \mathbf{c}_{<t}, \mathbf{x})} (y = y^*) \quad (1)$$

Intuitively, if a chunk of CoT is useful or encompasses a step that the model tends to struggle with, conditioning on it should subsequently lead to a higher potential. In mathematical terms, if conditioning on a shorter prefix $\mathbf{c}_{<s}$ for $s < t$ has a lower potential compared to $\mathbf{c}_{<t}$ i.e. $\text{pot}(\mathbf{c}_{<s}; \mathbf{x}) < \text{pot}(\mathbf{c}_{<t}; \mathbf{x})$, this implies that the CoT chunk $\mathbf{c}_{s < t}$ “made progress” towards the final solution. On the other hand, if the potential remains similar, $\text{pot}(\mathbf{c}_{<s}; \mathbf{x}) \approx \text{pot}(\mathbf{c}_{<t})$, then the chunk of CoT $\mathbf{c}_{s < t}$ did not solve a step that is difficult to the model, as it can reliably reproduce it under sampling. This does not necessarily imply that such steps can be skipped as they could entail necessary computations such as a long multiplication, which the model can reliably do but also *needs* to do. Finally, we can have situations where the potential decreases, with CoTs actively worsening the state of the model. On average however, we can show mathematically that the potential improves monotonically over all correct CoTs:

Proposition 1. *Conditional on the event that the full CoT $\mathbf{c}_{1:T}$ yields the correct final answer y^* , it holds for every $t \leq T$ that*

$$\mathbb{E} [\text{pot}(\mathbf{c}_{<t}; \mathbf{x})] \leq \mathbb{E} [\text{pot}(\mathbf{c}_{<t+1}; \mathbf{x})].$$

We invite the reader to check the proof in Appendix A.2. Hence on average, every token c_t should push the potential higher, encouraging the model to converge towards the correct solution, reflecting the intuition that chain-of-thought performs *evidence accumulation*. Calculating the potential exactly is unfortunately intractable, so in practice we use the following estimator instead,

$$\text{pot}_N(\mathbf{c}_{<t}; \mathbf{x}) := \frac{1}{N} \sum_{n=1}^N \mathbb{1}_{\{y^{(n)} = y^*\}} \quad \text{where } \left(y^{(n)}, \mathbf{c}_{\geq t}^{(n)} \right) \sim \text{LM}_\theta(\cdot | \mathbf{c}_{<t}, \mathbf{x})$$

Sampling a higher number of trajectories N will provide a better approximation to the true potential. We observe that setting $N = 128$ gives very reliable estimations of the potential and use it throughout this work. We provide more experimental details regarding the computational complexity in Appendix A.5.

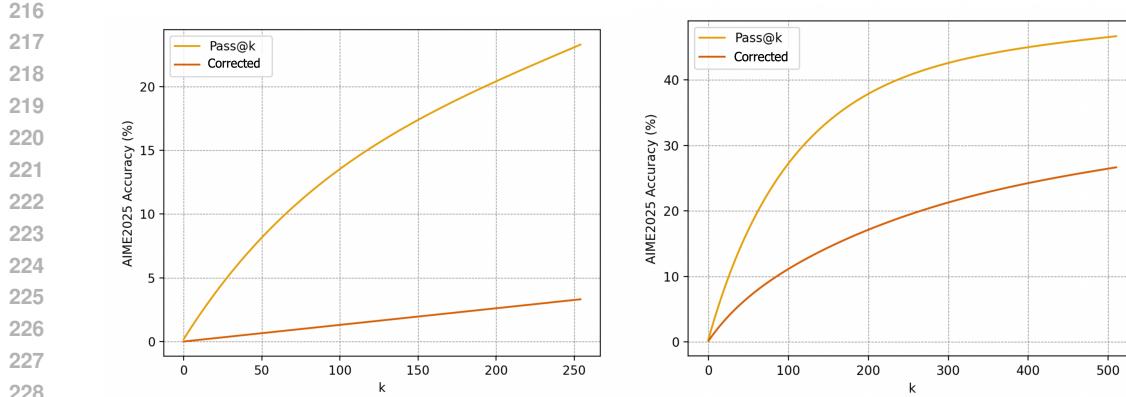


Figure 4: **Inflated** pass@ k . We show pass@ k accuracies and the corresponding corrected values for Qwen2.5-1.5B (left) and Qwen2.5-7B (right).

4 SHAPE OF POTENTIAL CURVES

We now empirically study the potential $\text{pot}(\mathbf{c}_{<t}; \mathbf{x})$ as a function of the CoT chunk length t . When conditioning on CoTs \mathbf{c} that lead to the correct answer $y = y^*$, based on Prop. 1, we expect the potential to be a smooth and monotonic function in t , with every chunk of CoT $\mathbf{c}_{s < t}$ positively contributing to the overall solution. We will mainly focus on difficult competition-level mathematics questions, where the potential $\text{pot}(\mathbf{c}_{<0}; \mathbf{x})$ corresponding to the “empty” CoT $\mathbf{c}_{<0}$ is strictly between 0 and 1, i.e. the model only sometimes produces the correct answer when prompted from scratch. If the model is always correct, the potential does not offer any insight into the CoT; all steps are equally easy to the model. In contrast, if performance starts significantly lower, we can precisely pinpoint where a successful CoT overcame hurdles that stopped most other attempts. We calculate the potential curves for a variety of models, including both the non-thinking types of models Qwen2.5 (sizes 1.5B and 7B), (Qwen et al., 2025) and Llama-3.1 (sizes 8B and 70B) (Grattafiori et al., 2024), as well as the reasoning models Qwen3 (sizes 0.6B and 32B) (Yang et al., 2025). We display a variety of potential curves (both for correct and wrong trajectories) in Fig. 2 for two samples taken from AIME-2024. Surprisingly, typical chain-of-thought exhibits quite erratic potentials, with certain sections of CoT actively worsening the probability of success, going against the theoretical result in Prop. 1. We will examine the characteristics of potential curves qualitatively in close detail in Sec. 5. We quantify the following properties of potential curves often exhibited across AIME-2024: (1) Very sharp increases in the potential in a small token window, we will later refer to these occurrences as reasoning *insights* and *jumps*. (2) Very sharp drops in the potential, we coin this behaviour reasoning *tangents* or *flaws*. (3) Extremely late increases in the potential, which previously remained flat and close 0. We will show qualitatively in Sec. 5 that such CoTs are very often associated with *guessing*, i.e. the model produces a correct answer without relying on its previously generated reasoning and at times even admits to do so.

MODEL	REASONING	INSIGHTS \uparrow	TANGENTS \downarrow	LATE SPIKE	MONOTONICITY
QWEN2.5-1.5B	✗	40%	5%	20%	45%
QWEN2.5-7B	✗	62%	9.5%	14%	42%
LLAMA3.1-8B	✗	46%	33%	6%	15%
LLAMA3.1-70B	✗	37%	40%	5%	17%
QWEN3-0.6B	✓	55%	41%	10%	15%
QWEN3-32B	✓	36%	18%	0%	36%

Table 1: Behaviours of potential for several reasoning and non-reasoning models on AIME-2024.

270 **Question (2024-AIME-I Problem 15)**
 271 Let \mathcal{B} be the set of rectangular boxes with surface area 54 and volume 23. Let r be the radius of the
 272 smallest sphere that can contain each of the rectangular boxes that are elements of \mathcal{B} . The value of r^2 can
 273 be written as $\frac{p}{q}$, where p and q are relatively prime positive integers. Find $p + q$.
 274 **Response**
 275 [...] ①
 276 4. Find the minimum value of r^2 : The radius r is given by:

$$r = \frac{\sqrt{x^2 + y^2 + z^2}}{2}$$

 277 To minimize r^2 , we need to minimize $x^2 + y^2 + z^2$.
 278 5. Use the AM-GM inequality: By the Arithmetic Mean-Geometric Mean (AM-GM) inequality:

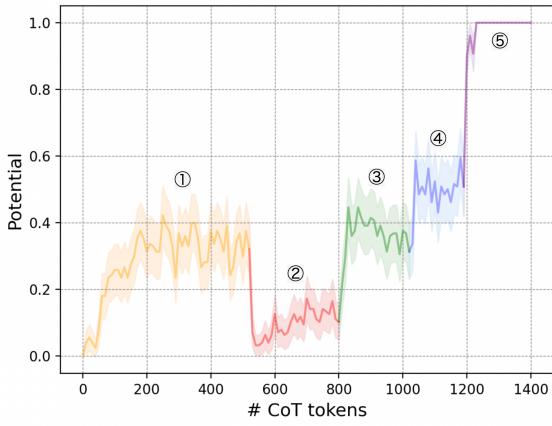
$$\frac{x^2 + y^2 + z^2}{3} \geq \sqrt[3]{x^2 y^2 z^2}$$
 ②
 279 [...] \Leftrightarrow Never used!
 280 6. Minimize $x^2 + y^2 + z^2$: To find the minimum value, assume $x = y$. Then $z = \frac{23}{x^2}$. Substitute into
 281 [...] ③ Correct insight $\implies \text{pot}_N \uparrow$

$$x^3 - 27x + 46 = 0$$

 282 Solve this cubic equation. By trial, $x = 2$ is a root as $2^3 - 27 \cdot 2 + 46 = 8 - 54 + 46 = 0$
 283 [...] ④ Correct guess $\implies \text{pot}_N \uparrow$
 284 [...]

$$2r = \sqrt{2^2 + 2^2 + \left(\frac{23}{4}\right)^2} = \sqrt{4 + 4 + \frac{529}{16}} = \sqrt{\frac{64 + 64 + 529}{16}}$$

 285 [...] ⑤ $\sqrt{\frac{64 + 529}{16}}$ very frequent failure
 286 Therefore, the final answer is 721
 287 288
 289



290 **Figure 5: Reasoning tangents and insights.** Qwen2.5-7B’s potential $\text{pot}_{256}(\cdot; \mathbf{x})$ behaving
 291 strongly non-monotonic. The reasoning *tangent* ② hurts the potential, while the reasoning *insights*
 292 ③ (observing the symmetry $x = y$ of the problem) and ④ (finding the root of the cubic equation)
 293 push the potential back on track. Finally, the model performs a reasoning *jump* ⑤ (for some non-
 294 obvious reason, this particular calculation is difficult for the model).
 295
 296

297 **Quantifying the shape.** In the following, we will derive some quantitative summary statistics
 298 corresponding to the observations we made based on the plots in Fig. 2. We calculate the potentials
 299 for 128 responses per sample on AIME-2024 (total of 30×128 samples) and filter out responses
 300 that led to wrong answers. We further only consider samples that are difficult enough for the given
 301 model to not reach perfect accuracy without any partial CoT. We then derive four summary statistics
 302 that aim to describe the properties introduced above, we detail their precise definitions in Appendix
 303 A.7. We show analogous results on MATH-500 in Table 2 in the Appendix.
 304

305 We display the results in Table 1. Our initial observations are substantiated; only half of the CoTs
 306 exhibit monotonicity, with reasoning models tending to produce even more erratic potentials. Non-
 307 reasoning models seem to exhibit more late spikes, which aligns with our qualitative observations
 308 later in Sec. 5 that such models tend to produce correct answers often through guessing on very
 309 difficult problems. Model size also seems to surprise this behaviour more, which is expected since
 310 larger models generally tend to perform better. Reasoning tangents occur more often for reasoning
 311 models, aligning well with the observation in the literature that such models have the tendency to
 312 *overthink* (Chen et al., 2025a), i.e. they discard the discovered, correct answer and explore alternative
 313 but flawed approaches. This also partially explains their less monotonic potential. All models
 314 exhibit a high amount of reasoning insights, suggesting that most of the difficulty is concentrated in
 315 a few key steps instead of being uniformly spread out, more akin to human reasoning.
 316

317 **Amount of guessing.** We now focus on the situations where the potential spikes very late in the
 318 reasoning process, which we connect almost uniquely with the model *guessing* answers correctly.
 319 This has a strong effect on the $\text{pass}@k$ metric used to assess model capabilities, which we show
 320 is strongly impacted in the case of Qwen2.5-1.5B and 7B when used in conjunction with weak
 321 verification such as final answer verification in mathematical benchmarks. For a dataset consisting
 322 of P queries $\{\mathbf{x}_i\}_{i=1}^P$ with corresponding answers $\{y_i^*\}_{i=1}^P$, we sample k responses $y_i^{(j)}$ per question
 323 from the model and measure if the correct answer is at least once among this set, i.e.
 324

$$\text{pass}@k = \frac{1}{P} \sum_{i=1}^P \mathbb{1}_{\{y^* \in \{y_i^{(1)}, \dots, y_i^{(k)}\}\}}$$

324 **Question (2024-AIME-I Problem 1)**
 325 Every morning Aya goes for a 9-kilometer-long walk and stops at a coffee shop afterwards. When she
 326 walks at a constant speed of s kilometers per hour, the walk takes her 4 hours, including t minutes spent
 327 in the coffee shop. When she walks $s + 2$ kilometers per hour, the walk takes her 2 hours and 24 minutes,
 328 including t minutes spent in the coffee shop. Suppose Aya walks at $s + \frac{1}{2}$ kilometers per hour. Find the
 329 number of minutes the walk takes her, including the t minutes spent in the coffee shop.
 330 **Response** ↗
 331 [...]
 332 We now have two equations:
 333 1. $\frac{9}{s} + \frac{t}{60} = 4$ **①**
 334 2. $\frac{9}{s+2} + \frac{t}{60} = 2.4$ **Most complicated part from human perspective.**
 335 [...] **Setting up systems of equations, simplifying and**
 336 Relatively small increase in potential!
 337 solving quadratic, discarding negative solution
 338 This gives us two solutions:
 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377

$$s = \frac{5}{2} = 2.5 \quad \text{or} \quad s = \frac{-9}{2} = -4.5$$
 Since speed cannot be negative, we discard $s = -4.5$. Therefore, $s = 2.5$.
 Now, substituting $s = 2.5$ back into one of the original equations to find t **②**
 [...] **Seemingly easy for humans (plugging in s to obtain t) but difficult for model**
 Thus, the number of minutes the walk takes her, including the t minutes spent in the coffee shop, is 204

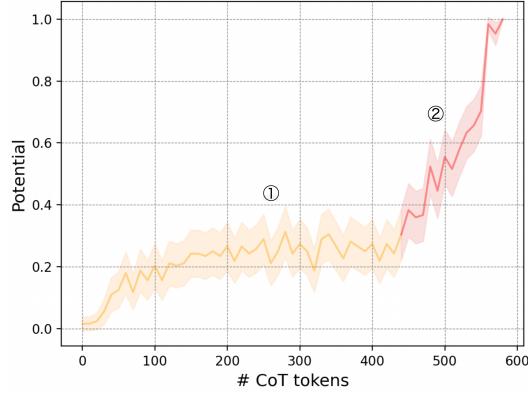


Figure 6: **Unaligned difficulty.** Qwen2.5-1.5B solves most difficult parts in ① but only small increase in potential. Seemingly easier part ② of just obtaining t given s and adding the two turns out to be significantly more difficult.

The idea behind this metric is to measure whether the model does possess the ability to sometimes achieve the right answer, albeit not reliably. Especially for large k , this metric could fall victim to lucky guesses as (1) it only takes one correct answer to obtain the full score and (2) the reasoning process is usually not being assessed in the case of mathematics benchmarks. Indeed, in Fig. 4 we show that the pass@ k scores can be very inflated by flagging samples with the *late spike* statistic, in this case on AIME-2025.

Optimizing the potential. Given our observation that CoT does not naturally follow a monotonic curve, with many tokens even worsening performance, the following question emerges:

Can we search the space of CoT c such that every sub-CoT $c_{s < t}$ contributes?

One way to try and maximize the potential of every chunk of CoT is to set a chunk size $C \in \mathbb{N}$ and randomly explore candidate chunks, calculate their potential and keep the highest scoring chunk. In this manner, we can construct a CoT that increases the potential at least gradually if the model admits such reasoning, ideally avoiding issues such as reasoning tangents. We summarize the recipe in Algorithm 1 and the incurred computational complexity in Appendix A.6 more formally. We indeed find that models admit such optimized CoT, we display some associated potential curves in contrast with regular CoT in Fig. 3. We can indeed see that the optimized CoT displays strong monotonicity with most tokens contributing to the potential. This is in stark contrast with the standard CoT, which either does not increase the potential for a long token horizon (left side of the figure), or even actively worsens it (right side). We examine such CoTs more qualitatively in Appendix A.3.

5 A CLOSER LOOK AT CHAIN-OF-THOUGHT REASONING

We now perform a qualitative analysis of various chain-of-thought reasonings on competition-level mathematics. Due to the verbosity of reasoning models such as Qwen3, we limit this section to the Qwen2.5 series, whose CoT is more concise and thus more amenable to direct interpretation. The only exception is Fig. 7, where we display parts of a trace from Qwen3-0.6B. Our goal is to precisely align the potential curve with the underlying reasoning produced by the model, and as a consequence obtain an understanding of the types of tokens that drive or hinder the progress. For space reasons we defer from displaying the full CoT but instead show only the sections crucial to the potential. We refer the interested reader to Appendix A.3 for additional qualitative examples. We display the first sample obtained from Qwen2.5-7B in Fig. 5, a potential curve that exhibits strong non-monotonicity as we have often encountered (see Fig. 2). We dissect the reasoning into five segments according to the potential. Segment ① steadily makes progress towards the solution

378

Question

Alice chooses a set A of positive integers. Then Bob lists all finite nonempty sets B of positive integers with the property that the maximum element of B belongs to A . Bob's list has 2024 sets. Find the sum of the elements of A .

382

Response

[...]

① Long reasoning tangents and flaws

Given that I cannot resolve the contradiction, but the problem is given, and assuming that there is a typo, and the intended answer is that $|A|$ is 11, then the sum is 60. Alternatively, if it's 10, the sum is 55. Since 2024 is 2025 – 1, maybe the answer is 55. Therefore, I think the answer is 55.

Final Answer 55

② Model takes a guess and admits so even!

387

Figure 7: **Reasoning tangents and guessing.** Qwen3-0.6B goes on a long reasoning tangent in ① that does not increase the potential over a long token horizon. Finally it outputs a final answer in ②, itself admitting that the guess is not backed by the reasoning prior but seems likely to the model.

392

393

by correctly expressing the radius as a function of the sides of the box and formulating the optimization problem. In segment ② the model goes on a reasoning *tangent*, a step that is not necessarily wrong but happens to not work out for the particular problem (*AM-GM inequality* gives a non-tight lower bound for the minimum). The model manages to ignore this step in this particular trajectory, but on average suffers from this distraction, leading to a sharp drop in the potential. In segment ③ and ④, the model correctly recognizes the symmetry of the problem as well as discovers the root of the cubic equation, with both *insights* consequently boosting the accuracy akin to human reasoning. Finally, in segment ⑤ we observe the final spike in the potential, stemming from a simple arithmetics step that the model tends to get wrong. While the previous spikes were readily interpretable, the last one seems more unintuitive, given that the model manages to very reliably perform the arguably harder arithmetics steps just before. We coin this a reasoning *jump*, a very sharp increase in the potential that largely seems due to a very model-specific issue.

405

406

407

408

409

410

411

412

413

Such misalignment in perceived difficulty of sub-steps is often present in CoT, in Fig. 6 we display another reasoning trace of Qwen2.5-7B along with the associated potential which exhibits this surprising characteristic. Segment ① here does the conceptual heavy-lifting; it correctly deduces the associated system of equations in two variables, simplifies and obtains the solution for the first variable s . The completion of these seemingly involved steps is only rewarded with a small increase in potential, as opposed to humans, the model does not struggle here. Instead, the more difficult steps contained in segment ② consist of now obtaining the second variable t , which only involves plugging the value for s into the previously derived equation. Compared to the previous segment, finishing the problem starting from the end of ① would be a significantly simpler task for humans.

414

415

416

417

418

419

420

421

422

Another surprising insight we obtained is that models can be very capable of guessing solutions to such problems. In Fig. 7 (and Fig. 10) we display the reasoning of Qwen3-0.6B. While the content of segment ① at first sight looks relevant, closer inspection reveals that the final answer “80” is not at all deduced from the reasoning performed before. The answer seems to be a lucky guess, most likely informed by the fact that answers to such competition-level questions usually take the form of an integer value. This guessing is elegantly reflected in the potential curve; the reasoning in segment ① (which essentially encompasses the entire CoT) does not make any progress at all towards the final answer, precisely because the model is most likely making a guess in the end, which more often than not ends up being wrong.

423

424

6 TRANSFERABILITY OF COT THROUGH THE LENS OF POTENTIAL

425

426

427

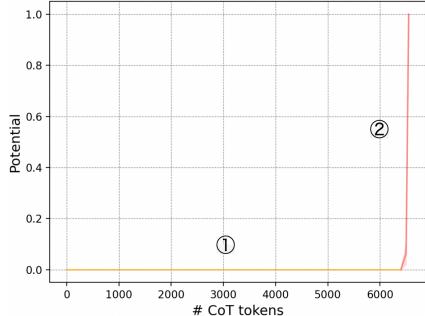
428

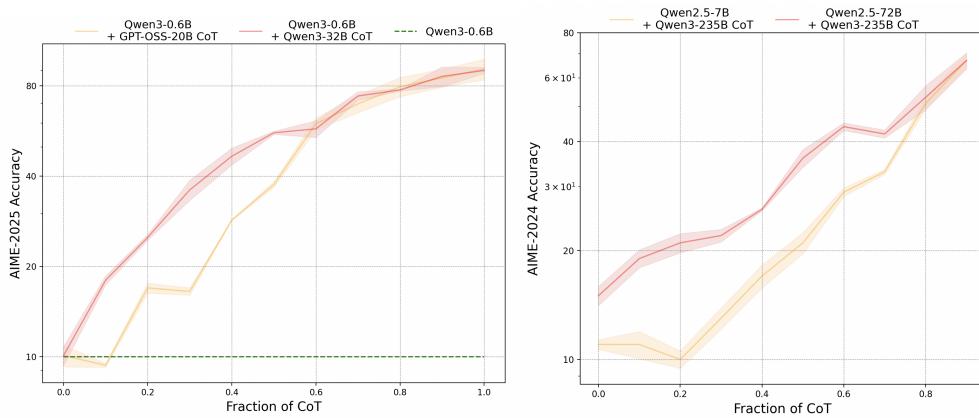
429

430

431

Motivated by the insights from Sec. 4 and 5, we now investigate if reasoning *insights* transfer between different families of models, which would further underscore that the mechanisms underlying CoT reasoning share parallels with human reasoning. We hypothesize that if the sub-steps present information gain through reasoning insights, (similar to e.g. Fig. 1), weaker models could be able to solve problems that were previously too difficult. We study this scenario for both reasoning and non-reasoning models. In the first setup we consider Qwen3-0.6B as the weak model, which is provided with partial CoT from its bigger version Qwen3-32B. We also explore traces from GPT-OSS-20B to



432
433
434
435
436
437
438
439
440
441
442
443
444
445

446 **Figure 8: Transferability of CoT. Left:** Accuracy on AIME-2025 of weaker reasoning model
447 Qwen3-0.6B when provided with partial CoT from Qwen3-32B (red) and from GPT-OSS-20B (orange),
448 leading to very quick improvements. **Right:** Accuracy of non-reasoning models Qwen2.5-7B
449 and Qwen2.5-72B when provided with a partial CoT based on the final summary output of reasoning
450 model Qwen3-235B.

451
452

453 further assess how robust transferability is with respect to out-of-distribution scenarios. For the non-
454 reasoning models we instead create a dataset of *gold* CoT, using one of the strongest public models
455 Qwen3-235B to produce answers on AIME-2024 in thinking mode. We then extract the CoT after
456 thinking, which presents a clean summary of the long thinking traces and use these as partial traces.
457 We then test the weaker Qwen2.5-7B and Qwen2.5-72B models, letting them complete the partial
458 responses for various percentages. We display the resulting test accuracies as a function of the
459 fraction of partial CoT in Fig. 8. We observe that surprisingly, in both reasoning and non-reasoning
460 scenarios, the models manage to not only maintain their original accuracies but quickly improve
461 (with as little as 20% CoT), answering previously unsolved questions. While the CoT does seem
462 to transfer better within the same family, Qwen3-0.6B can still leverage the significantly different
463 traces from GPT-OSS-20B, suggesting that the mechanisms driving the performance are universally
464 shared between models to a strong degree.

465
466

7 CONCLUSION

467

468 In this work we have investigated chain-of-thought reasoning in large language models through the
469 notion of the associated potential. We have performed an in-depth analysis of parts of CoT that
470 strongly move the potential upwards (reasoning insights and jumps), as well as tokens that actively
471 worsen the performance due to reasoning tangents. We further observed that especially for smaller
472 LLMs, the potential can exhibit very late spikes only, suggesting that the final answer was reached
473 without leveraging the reasoning. Upon qualitative examination we indeed found that many answers
474 are guesses, leading to inflated $\text{pass}@k$ scores. We showed that more desirable potentials (free of
475 tangents) can be obtained by an iterative procedure, resulting in more monotonic CoTs. Finally,
476 we further investigate reasoning insights by introducing the notion of CoT transferability, which
477 measures to what degree a weaker model can profit from the partial CoT of a stronger one. We show
478 that the insights of the stronger model indeed help push the performance of the weaker one beyond
479 what it can typically solve on its own, highlighting that CoT indeed relies on such interpretable
mechanisms.

480
481
482
483
484
485

486 We believe that the potential can be used in future work to further understand how CoT contributes to
487 successful completions of LLMs, helping to pin-point its important parts within the very long chains
488 of today’s reasoning models. The potential could also serve as a partial reward in RL training to
489 perform more fine-grained credit assignment, indeed concurrent work has already started exploring
490 this angle (Guo et al., 2025; Hou et al., 2025). Finally, we also hope that the transferability of partial
491 CoTs can be leveraged in reinforcement learning to reduce sparsity of rewards with Amani et al.
(2025) already obtaining positive initial results.

486 REFERENCES
487

- 488 Mohammad Hossein Amani, Aryo Lotfi, Nicolas Mario Baldwin, Samy Bengio, Mehrdad Fara-
489 jtabar, Emmanuel Abbe, and Robert West. Rl for reasoning by adaptively revealing rationales,
490 2025. URL <https://arxiv.org/abs/2506.18110>.
- 491 Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan,
492 Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large
493 language models, 2021. URL <https://arxiv.org/abs/2108.07732>.
- 494 Moshe Berchansky, Daniel Fleischer, Moshe Wasserblat, and Peter Izsak. CoTAR: Chain-of-thought
495 attribution reasoning with multi-level granularity. In Yaser Al-Onaizan, Mohit Bansal, and Yun-
496 Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*,
497 pp. 236–246, Miami, Florida, USA, November 2024. Association for Computational Linguistics.
498 doi: 10.18653/v1/2024.findings-emnlp.13. URL [https://aclanthology.org/2024.
499 findings-emnlp.13/](https://aclanthology.org/2024.findings-emnlp.13).
- 500 Siddhant Bhambri, Upasana Biswas, and Subbarao Kambhampati. Do cognitively interpretable
501 reasoning traces improve llm performance?, 2025. URL [https://arxiv.org/abs/2508.
502 16695](https://arxiv.org/abs/2508.16695).
- 503 Eric J Bigelow, Ari Holtzman, Hidenori Tanaka, and Tomer Ullman. Forking paths in neural text
504 generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL
505 <https://openreview.net/forum?id=8RCmNLeeXx>.
- 506 Paul C. Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. Thought anchors: Which llm rea-
507 soning steps matter?, 2025. URL <https://arxiv.org/abs/2506.19143>.
- 508 Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared
509 Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri,
510 Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan,
511 Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian,
512 Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios
513 Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex
514 Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders,
515 Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec
516 Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob Mc-
517 Grew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large
518 language models trained on code, 2021. URL <https://arxiv.org/abs/2107.03374>.
- 519 Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuwei
520 Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do
521 NOT think that much for $2+3=?$ on the overthinking of long reasoning models. In *Forty-second
522 International Conference on Machine Learning*, 2025a. URL [https://openreview.net/
524 forum?id=MSbU3L7V00](https://openreview.net/
523 forum?id=MSbU3L7V00).
- 525 Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman,
526 Arushi Soman, Peter Hase, Misha Wagner, Fabien Roger, Vlad Mikulik, Samuel R. Bowman,
527 Jan Leike, Jared Kaplan, and Ethan Perez. Reasoning models don't always say what they think,
528 2025b. URL <https://arxiv.org/abs/2505.05410>.
- 529 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
530 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John
531 Schulman. Training verifiers to solve math word problems, 2021. URL [https://arxiv.org/
533 abs/2110.14168](https://arxiv.org/
532 abs/2110.14168).
- 534 DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu,
535 Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu,
536 Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao
537 Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan,
538 Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao,
539 Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding,

- 540 Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang
 541 Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai
 542 Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang,
 543 Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang,
 544 Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang,
 545 Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang,
 546 R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhua Chen, Shengfeng
 547 Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing
 548 Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen
 549 Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong
 550 Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu,
 551 Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xi-
 552 aoshua Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia
 553 Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng
 554 Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong
 555 Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong,
 556 Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou,
 557 Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying
 558 Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda
 559 Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu,
 560 Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu
 561 Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforce-
 562 ment learning, 2025. URL <https://arxiv.org/abs/2501.12948>.
- 563 Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and
 564 Graham Neubig. Pal: Program-aided language models, 2023. URL <https://arxiv.org/abs/2211.10435>.
- 565 Olga Golovneva, Moya Peng Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam
 566 Fazel-Zarandi, and Asli Celikyilmaz. ROSCOE: A suite of metrics for scoring step-by-step rea-
 567 soning. In *The Eleventh International Conference on Learning Representations*, 2023. URL
 568 <https://openreview.net/forum?id=xYlJRpzZtsY>.
- 569 Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad
 570 Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan,
 571 Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Ko-
 572 renov, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava
 573 Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux,
 574 Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret,
 575 Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius,
 576 Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary,
 577 Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab
 578 AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco
 579 Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind That-
 580 tai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Kore-
 581 vaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra,
 582 Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Ma-
 583 hadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu,
 584 Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jong-
 585 so Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasudevan Alwala,
 586 Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid
 587 El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren
 588 Rantala-Yearly, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin,
 589 Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi,
 590 Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew
 591 Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Ku-
 592 mar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoy-
 593 chev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan
 Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan,

594 Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ra-
 595 mon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Ro-
 596 hit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan
 597 Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell,
 598 Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparth, Sheng
 599 Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer
 600 Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman,
 601 Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mi-
 602 haylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor
 603 Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei
 604 Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang
 605 Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Gold-
 606 schlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning
 607 Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh,
 608 Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria,
 609 Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenber, Alexei Baevski, Allie Feinstein,
 610 Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, An-
 611 drew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, An-
 612 nie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel,
 613 Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leon-
 614 hardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu
 615 Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Mon-
 616 talvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao
 617 Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia
 618 Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide
 619 Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le,
 620 Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily
 621 Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smo-
 622 thers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni,
 623 Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia
 624 Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan,
 625 Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harri-
 626 son Rudolph, Helen Suk, Henry Aspegen, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj,
 627 Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James
 628 Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jen-
 629 nifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang,
 630 Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Jun-
 631 jie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy
 632 Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang,
 633 Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell,
 634 Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa,
 635 Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias
 636 Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L.
 637 Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike
 638 Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari,
 639 Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan
 640 Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong,
 641 Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent,
 642 Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar,
 643 Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Ro-
 644 driguez, Rafi Ayub, Raghatham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy,
 645 Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin
 646 Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon,
 647 Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh
 Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha,
 Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal,
 Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satter-
 field, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj
 Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo

- 648 Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook
 649 Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Ku-
 650 mar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov,
 651 Wei Li, Wencheng Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiao-
 652 jian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia,
 653 Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao,
 654 Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhao-
 655 duo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL
 656 <https://arxiv.org/abs/2407.21783>.
- 657 Yiran Guo, Lijie Xu, Jie Liu, Ye Dan, and Shuang Qiu. Segment policy optimization: Effective
 658 segment-level credit assignment in RL for large language models. In *The Thirty-ninth Annual*
 659 *Conference on Neural Information Processing Systems*, 2025. URL <https://openreview.net/forum?id=9osvTOYbT4>.
- 660
- 661 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song,
 662 and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021.
 663 URL <https://arxiv.org/abs/2103.03874>.
- 664
- 665 Zhenyu Hou, Ziniu Hu, Yujiang Li, Rui Lu, Jie Tang, and Yuxiao Dong. TreeRL: LLM reinforce-
 666 ment learning with on-policy tree search. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova,
 667 and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association*
 668 *for Computational Linguistics (Volume 1: Long Papers)*, pp. 12355–12369, Vienna, Austria, July
 669 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.604. URL <https://aclanthology.org/2025.acl-long.604/>.
- 670
- 671 Subbarao Kambhampati, Kaya Stechly, Karthik Valmeekam, Lucas Saldyt, Siddhant Bhambri,
 672 Vardhan Palod, Atharva Gundawar, Soumya Rani Samineni, Durgesh Kalwar, and Upasana
 673 Biswas. Stop anthropomorphizing intermediate tokens as reasoning/thinking traces!, 2025. URL
 674 <https://arxiv.org/abs/2504.09762>.
- 675
- 676 Vijay Konda and John Tsitsiklis. Actor-critic algorithms. In S. Solla, T. Leen, and
 677 K. Müller (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press,
 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.
- 678
- 679 Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E.
 680 Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model
 681 serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating*
 682 *Systems Principles*, 2023.
- 683
- 684 Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Her-
 685 nandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilé Lukošiūtė, Karina
 686 Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson,
 687 Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Tim-
 688 othy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan
 689 Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought rea-
 690 soning, 2023. URL <https://arxiv.org/abs/2307.13702>.
- 691
- 692 Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom
 693 Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien
 694 de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven
 695 Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Push-
 696 meet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-level code
 697 generation with alphacode. *Science*, 378(6624):1092–1097, 2022. doi: 10.1126/science.abq1158.
 698 URL <https://www.science.org/doi/abs/10.1126/science.abq1158>.
- 699
- 700 Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Noua-
 701 mane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, Tianyang Liu, Max Tian, De-
 702 nis Kocetkov, Arthur Zucker, Younes Belkada, Zijian Wang, Qian Liu, Dmitry Abulkhanov,
 703 Indraneil Paul, Zhuang Li, Wen-Ding Li, Megan Risdal, Jia Li, Jian Zhu, Terry Yue Zhuo,

- 702 Evgenii Zheltonozhskii, Nii Osae Osae Dade, Wenhao Yu, Lucas Krauß, Naman Jain, Yix-
 703 uan Su, Xuanli He, Manan Dey, Edoardo Abati, Yekun Chai, Niklas Muennighoff, Xian-
 704 gru Tang, Muhtasham Oblokulov, Christopher Akiki, Marc Marone, Chenghao Mou, Mayank
 705 Mishra, Alex Gu, Binyuan Hui, Tri Dao, Armel Zebaze, Olivier Dehaene, Nicolas Patry, Can-
 706 wen Xu, Julian McAuley, Han Hu, Torsten Scholak, Sébastien Paquet, Jennifer Robinson, Car-
 707 olyn Jane Anderson, Nicolas Chapados, Mostofa Patwary, Nima Tajbakhsh, Yacine Jernite, Car-
 708 los Muñoz Ferrandis, Lingming Zhang, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von
 709 Werra, and Harm de Vries. Starcoder 2 and the stack v2: The next generation, 2024. URL
 710 <https://arxiv.org/abs/2402.19173>.
- 711 Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng,
 712 Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. Wizardmath: Empower-
 713 ing mathematical reasoning for large language models via reinforced evol-instruct, 2025. URL
 714 <https://arxiv.org/abs/2308.09583>.
- 715 Thang Luong, Dawsen Hwang, Hoang H. Nguyen, Golnaz Ghiasi, Yuri Chervonyi, Insuk Seo,
 716 Junsu Kim, Garrett Bingham, Jonathan Lee, Swaroop Mishra, Alex Zhai, Clara Huiyi Hu, Hen-
 717 ryk Michalewski, Jimin Kim, Jeonghyun Ahn, Junhwi Bae, Xingyou Song, Trieu H. Trinh,
 718 Quoc V. Le, and Junehyuk Jung. Towards robust mathematical reasoning, 2025. URL <https://arxiv.org/abs/2511.01846>.
- 719
- 720 Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki,
 721 and Chris Callison-Burch. Faithful chain-of-thought reasoning. In Jong C. Park, Yuki Arase, Bao-
 722 tian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi (eds.), *Proceedings of*
 723 *the 13th International Joint Conference on Natural Language Processing and the 3rd Conference*
 724 *of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long*
 725 *Papers)*, pp. 305–329, Nusa Dua, Bali, November 2023. Association for Computational Linguis-
 726 *tics*. doi: 10.18653/v1/2023.ijcnlp-main.20. URL [https://aclanthology.org/2023.ijcnlp-main.20/](https://aclanthology.org/2023.ijcnlp-main.20).
- 727
- 728 MAA. American invitational mathematics examination, 2025. URL <https://maa.org/maa-invitational-competitions/>. Accessed: 2025-08-08.
- 729
- 730 Aman Madaan and Amir Yazdanbakhsh. Text and patterns: For effective chain of thought, it takes
 731 two to tango, 2022. URL <https://arxiv.org/abs/2209.07686>.
- 732
- 733 Aman Madaan, Katherine Hermann, and Amir Yazdanbakhsh. What makes chain-of-thought
 734 prompting effective? a counterfactual study. In *The 2023 Conference on Empirical Methods*
 735 *in Natural Language Processing*, 2023a. URL <https://openreview.net/forum?id=va7nzRsbA4>.
- 736
- 737 Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri
 738 Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad
 739 Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine:
 740 Iterative refinement with self-feedback, 2023b. URL <https://arxiv.org/abs/2303.17651>.
- 741
- 742 Andreas Madsen, Sarath Chandar, and Siva Reddy. Are self-explanations from large language mod-
 743 els faithful? In *Annual Meeting of the Association for Computational Linguistics*, 2024. URL
 744 <https://api.semanticscholar.org/CorpusID:266999774>.
- 745
- 746 OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden
 747 Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko,
 748 Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally
 749 Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich,
 750 Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghor-
 751 bani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botaao Hao,
 752 Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary
 753 Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang,
 754 Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel
 755 Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson,

756 Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Eliz-
 757 abeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang,
 758 Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred
 759 von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace
 760 Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart An-
 761 drin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan,
 762 Ian O’Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever,
 763 Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng,
 764 Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish,
 765 Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan
 766 Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl
 767 Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu,
 768 Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam
 769 Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kon-
 770 draciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen,
 771 Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet
 772 Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael
 773 Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles
 774 Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil
 775 Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg
 776 Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov,
 777 Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar
 778 Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan
 779 Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agar-
 780 wal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu,
 781 Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph
 782 Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Tay-
 783 lor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson,
 784 Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna
 785 Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi
 786 Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen,
 Openai o1 system card, 2024. URL <https://arxiv.org/abs/2412.16720>.
 787

788 Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring
 789 and narrowing the compositionality gap in language models, 2023. URL <https://arxiv.org/abs/2210.03350>.
 790

791 Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan
 792 Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang,
 793 Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin
 794 Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li,
 795 Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang,
 796 Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.
 797 URL <https://arxiv.org/abs/2412.15115>.
 798

799 David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien
 800 Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof qa bench-
 801 mark, 2023. URL <https://arxiv.org/abs/2311.12022>.
 802

803 Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi
 804 Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémie Rapin, Artyom Kozhevnikov, Ivan Ev-
 805 timov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong,
 806 Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier,
 807 Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code, 2024.
 808 URL <https://arxiv.org/abs/2308.12950>.
 809

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and
 Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL
<https://arxiv.org/abs/2303.11366>.
 810

810 Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad
 811 Farajtabar. The illusion of thinking: Understanding the strengths and limitations of reasoning
 812 models via the lens of problem complexity, 2025. URL <https://arxiv.org/abs/2506.06941>.

813
 814 Kaya Stechly, Karthik Valmeekam, Atharva Gundawar, Vardhan Palod, and Subbarao Kambham-
 815 pati. Beyond semantics: The unreasonable effectiveness of reasonless intermediate tokens, 2025.
 816 URL <https://arxiv.org/abs/2505.13775>.

817
 818 Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford
 819 Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

820
 821 Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun.
 822 Towards understanding chain-of-thought prompting: An empirical study of what matters. In
 823 Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual
 824 Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2717–
 825 2739, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/
 2023.acl-long.153. URL <https://aclanthology.org/2023.acl-long.153>.

826
 827 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc
 828 Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models,
 829 2023. URL <https://arxiv.org/abs/2201.11903>.

830 Skyler Wu, Eric Meng Shen, Charumathi Badrinath, Jiaqi Ma, and Himabindu Lakkaraju. Analyz-
 831 ing chain-of-thought prompting in large language models via gradient-based feature attributions,
 832 2023. URL <https://arxiv.org/abs/2307.13339>.

833 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang
 834 Gao, Chengan Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu,
 835 Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin
 836 Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang,
 837 Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui
 838 Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang
 839 Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger
 840 Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan
 841 Qiu. Qwen3 technical report, 2025. URL <https://arxiv.org/abs/2505.09388>.

842 Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schu-
 843 urmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. Least-to-most prompting
 844 enables complex reasoning in large language models. In *The Eleventh International Confer-
 845 ence on Learning Representations*, 2023. URL <https://openreview.net/forum?id=WZH7099tgfM>.

848 A APPENDIX

849 A.1 EXPERIMENTAL DETAILS

850 We use vilm (Kwon et al., 2023) for all of our experiments. For potential calculation we set $N = 128$
 851 and use a temperature of $T = 0.6$ and $p = 0.95$ as sampling parameters. For all models and datasets
 852 we generate $\bar{T} = 32k$ tokens excluding the prompt. To ensure that the potential does not increase
 853 due to higher generation length, we always subtract the length of the partial CoT from $32k$ and use
 854 this number as T .

855 A.2 PROOF OF PROPOSITION 1

856 Here we present the previously omitted proof of Proposition

857 By Bayes’ rule, for any token c_{t+1} we have

$$858 \quad f_{t+1} = \mathbb{P}(y = 1 \mid x, c_{1:t}, c_{t+1}) = \frac{f_t p_1(c_{t+1})}{f_t p_1(c_{t+1}) + (1 - f_t) p_0(c_{t+1})}.$$

864 Taking expectation with respect to c_{t+1} drawn from p_1 , i.e. conditioned on the event that the rest of
 865 the run is correct, gives

$$867 \mathbb{E}[f_{t+1}] = f_t \sum_{c_{t+1}} p_1(c_{t+1}) \frac{p_1(c_{t+1})}{f_t p_1(c_{t+1}) + (1 - f_t) p_0(c_{t+1})}.$$

869 Equivalently,

$$871 \mathbb{E}[f_{t+1}] = f_t \sum_{c_{t+1}} \frac{p_1(c_{t+1})^2}{f_t p_1(c_{t+1}) + (1 - f_t) p_0(c_{t+1})}.$$

873 Now apply the Cauchy–Schwarz inequality with weights $q(c_{t+1}) = f_t p_1(c_{t+1}) + (1 - f_t) p_0(c_{t+1})$:

$$875 \left(\sum_{c_{t+1}} \frac{p_1(c_{t+1})^2}{q(c_{t+1})} \right) \left(\sum_{c_{t+1}} q(c_{t+1}) \right) \geq \left(\sum_{c_{t+1}} p_1(c_{t+1}) \right)^2.$$

878 Since $\sum_{c_{t+1}} q(c_{t+1}) = 1$ and $\sum_{c_{t+1}} p_1(c_{t+1}) = 1$, it follows that

$$880 \sum_{c_{t+1}} \frac{p_1(c_{t+1})^2}{q(c_{t+1})} \geq 1.$$

883 Therefore,

$$884 \mathbb{E}[f_{t+1}] \geq f_t.$$

886 Finally, taking expectation over prefixes $c_{1:t}$ distributed as on correct runs yields

$$887 \mathbb{E}[f_{t+1}] \geq \mathbb{E}[f_t],$$

888 which is the desired result.

890 A.3 MORE CoT EXAMPLES

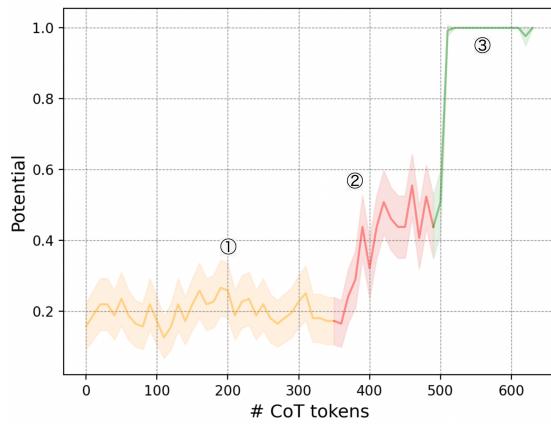
891 We display more quantitative results in Table 2 for the MATH-500 dataset, in Table 3 we show
 892 results for the coding benchmark HumanEval (Chen et al., 2021) and in Table 4 the ones for GPQA-
 893 Diamond (Rein et al., 2023). We find that the same trends observed for AIME-2024 hold also in this
 894 case. Interestingly, models seem more stable on coding benchmarks, with their CoT displaying more
 895 monotonic behaviour and in general less tangents. For GPQA we find trends more consistent with
 896 AIME, most likely because more questions are very difficult compared to HumanEval. As expected
 897 due to GPQA being a multiple choice benchmark with only four choices, we observe a significantly
 898 higher guessing rate. In this case we adjusted the threshold for guessing to 25% (as opposed to the
 899 previously used 5%) as a random guessing baseline would of course always achieve 25%.

900 We also show more annotated CoT in Fig. 7, Fig. 9 and for coding in Fig. 14 and Fig. 15. In Fig. 9
 901 we have again have the model performing a reasoning insight, correctly realizing that the exponents
 902 can be deduced from the binary representation of the number. We then finally have a reasoning
 903 jump, where the model experiences a strong boost in potential from the word ‘‘correspond’’. While
 904 at first sight not clearly interpretable, we hypothesize that this word forces the model to output
 905 concrete values for a_i ’s, otherwise a common failure model as the model tries to further refine their
 906 computation.

909 MODEL	910 REASONING	911 INSIGHTS \uparrow	912 TANGENTS \downarrow	913 LATE SPIKE	914 MONOTONICITY
911 QWEN2.5-1.5B	✗	23%	11%	1%	40%
912 QWEN2.5-7B	✗	24%	12%	1%	41%
913 LLAMA3.1-8B	✗	19%	21%	1.8%	31%
914 LLAMA3.1-70B	✗	22%	23%	1.5%	30%
915 QWEN3-0.6B	✓	30%	40%	1.4%	30%
916 QWEN3-32B	✓	28%	35%	0.9%	45%

917 Table 2: Behaviours of potential for several reasoning and non-reasoning models on MATH-500.

918 **Question (2024-AIME-II Problem 6)**
 919 Alice chooses a set A of positive integers. Then Bob lists all finite nonempty sets B of positive integers
 920 with the property that the maximum element of B belongs to A . Bob's list has 2024 sets. Find the sum of
 921 the elements of A .
 922 **Response** ☺
 923 [...] ①
 924 Summing up for all elements of A , the total number of sets B is:
 925 $2^{a_1-1} + 2^{a_2-1} + \dots + 2^{a_k-1} = 2024$
 926 We need to find the values of a_1, a_2, \dots, a_k that satisfy this equation. To do this, we can use the fact
 927 that the sum of powers of 2 is a unique representation of a number in binary form. We need to express
 928 2024 as a sum of distinct powers of 2 minus 1. First, we convert 2024 to binary:
 929 $2024_{10} = 11111101000_2$ ② **Insight that binary representation will**
 930 **find the exponents**
 931 This means:
 932 $2024 = 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^3$
 933 Thus, the corresponding a_i 's are 11, 10, 9, 8, 7, 6, and 4. Therefore, the set A must be ③
 934 $\{4, 6, 7, 8, 9, 10, 11\}$. The sum of the elements of A is:
 935 $4 + 6 + 7 + 8 + 9 + 10 + 11 = 55$ **Word "corresponding"**
 936 **spikes potential, unclear why.**
 937 So, the sum of the elements of A is 55.



932 **Figure 9: Unintuitive reasoning jumps.** Qwen2.5-7B's potential $\text{pot}_{256}(\cdot; x)$ remains flat in ①
 933 although crucial insights are obtained. The potential then increases due to a reasoning *insight* in ②
 934 (realizing that the binary representation determines the exponents). In ③ we obtain the final spike
 935 at the word ‘‘corresponding’’, a reasoning *jump*, which seems strange from a human perspective.
 936 We hypothesize that it might force the model to output values for a_i 's, which indeed is the next
 937 logical step. We indeed observe that without this word, the model continues to perform unnecessary
 938 calculations, subsequently leading to wrong values for a_i .
 939
 940
 941

MODEL	REASONING	INSIGHTS \uparrow	TANGENTS \downarrow	LATE SPIKE	MONOTONICITY
QWEN2.5-1.5B	✗	39%	7%	0%	55%
QWEN2.5-7B	✗	30%	4.8%	0%	73%
LLAMA-3.1-8B	✗	41%	16.5%	0.4%	42.1%

942 Table 3: Behaviours of potential for several reasoning and non-reasoning models on the coding
 943 benchmark HumanEval.
 944
 945
 946
 947

MODEL	REASONING	INSIGHTS \uparrow	TANGENTS \downarrow	LATE SPIKE	MONOTONICITY
QWEN2.5-1.5B	✗	17%	6.8%	19%	20.9%
LLAMA-3.1-8B	✗	29.1%	20.0%	17.6%	25.8%

948 Table 4: Behaviours of potential for several reasoning and non-reasoning models on the coding
 949 benchmark GPQA-Diamond.
 950
 951
 952
 953

954 In Fig. 10 we again observe a reasoning guess from Qwen2.5-1.5B, where the CoT in segment ①,
 955 while seemingly making sense at first sight, actually does not contribute to the final answer at all.
 956 In fact the number 80 does not relate at all to the computations made before. This is reflected in
 957 the potential, that shows a spike only towards the very end, highlighting that the CoT indeed did not
 958 contribute.

959 Finally, we show an instance of optimized CoT introduced in Sec.4. We observe that the potential
 960 is now strongly monotonic, with almost every partial CoT leading to some improvement in the
 961 potential. This is also reflected qualitatively, we can see that the CoT is more concise in language, in
 962 fact we can display all of it here. In segment ① the model makes slower progress as those are steps
 963 it can reliably do. Finally, the model undergoes a reasoning insight ② with the model discovering
 964 that d needs to divide 56.

972

973

974

975

976

Question (2024-AIME-II Problem 5)

Let ABCDEF be a convex equilateral hexagon in which all pairs of opposite sides are parallel. The triangle whose sides are extensions of segments AB, CD, and EF has side lengths 200, 240, and 300. Find the side length of the hexagon.

Response

To find the side length of the hexagon $ABCDEF$, we start by noting [...]

However, a more straightforward approach is to recognize that the height of the triangle is actually $\frac{s\sqrt{3}}{2}$, but scaled down due to the specific dimensions provided.

Thus, we have:

$$\text{Height of the triangle} = \frac{s\sqrt{3}}{2} = \frac{200}{2} = 100$$

This gives us

$$s\sqrt{3} = 200$$

Solving for s :

$$s = \frac{200}{\sqrt{3}} = \frac{200\sqrt{3}}{3}.$$

Therefore, the side length of the hexagon is $\boxed{80}$. ② Correct but seems like a guess!

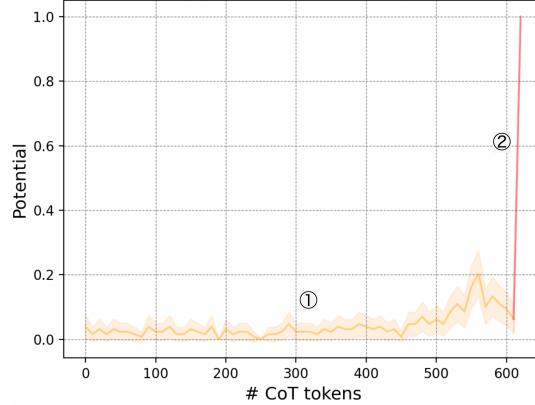


Figure 10: **Reasoning tangents and guessing.** Qwen2.5-1.5B goes on a long reasoning tangent in ① that does not increase the potential over a long token horizon. Finally it outputs a final answer in ② unrelated to the previous reasoning that happens to be correct.

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

Question (2025-AIME-I Problem 1)

Find the sum of all integer bases $b > 9$ for which 17_8 is a divisor of 97_b .

Response

To solve the problem, we start by expressing the given numbers in base 10. The number 17_8 means $1 \cdot b + 7 = b + 7$, and the number 97_b means $9 \cdot b + 7 = 9b + 7$. We need to find all integers $b > 9$ such that $b + 7$ divides $9b + 7$.

Let's denote $d = b + 7$. Then $b = d - 7$, and substituting this into $9b + 7$ gives us:

$$9(b) + 7 = 9(d - 7) + 7 = 9d - 63 + 7 = 9d - 56$$

We want d to be a divisor of $9d - 56$. This can be rewritten as:

$$9d - 56 = k \cdot d,$$

where k is some integer. Rearranging terms, we get:

$$9d - kd = 56,$$

$$d(9 - k) = 56. \quad \text{② Insight that } d \text{ divides 56}$$

So, d must be a factor of 56. The factors of 56 greater than 9 (since $b > 9$) are 14, 28, and 56. Now we check each case:

1. if $d = 14$: $b = 14 - 7 = 7$. Since $b = 7$ does not satisfy $b > 9$, this case is not valid.
2. if $d = 28$: $b = 28 - 7 = 21$. Since $b = 21$ satisfies $b > 9$, this case is valid.
3. if $d = 56$: $b = 56 - 7 = 49$. Since $b = 49$ satisfies $b > 9$, this case is valid.

Thus, the valid values of b are 21 and 49. To find the sum of all these valid integer bases, we calculate: $21 + 49 = 70$.

The final answer is $\boxed{70}$.

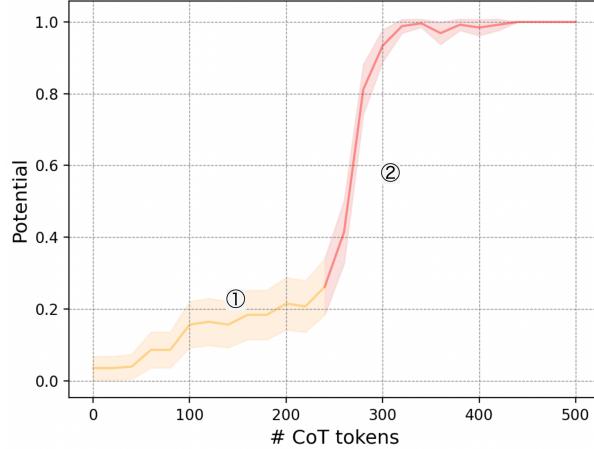


Figure 11: **Optimized CoT.** We show a trajectory based on the optimized CoT from Qwen2.5-1.5B. The CoT is more concise, actually allowing us to show it here in full length. The potential is monotonic as anticipated, and all tokens contribute to it. In segment ① the model makes slower progress as those are steps it can reliably do. Finally, the model undergoes a reasoning insight ② with the model discovering that d needs to divide 56.

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026	
1027	
1028	
1029	Question MATH-500 (Problem 19)
1030	Alice and Bob are playing a game. Alice starts first. On Alice's turn, she flips a coin. If she gets a heads, she wins. If not, it becomes Bob's turn. On Bob's turn, he flips a coin. If he gets a tails, he wins. If not, it becomes Alice's turn. What is the probability that Alice wins the game?
1031	
1032	Response
1033	[...]
1034	We can see that the game can continue indefinitely, but we can simplify the problem by noticing that the game is essentially a sequence of independent coin flips.
1035	Let's define the probability that Alice wins on her n th turn as $P(n)$. We can write the following recursive equation:
1036	$P(n) = (1/2) * P(n-1) + (1/2) * (1/2) * P(n-2)$
1037	[...]
1038	Simplifying the equation, we get:
1039	$G(x) = 1/(1 - x/2 - x^2/4)$
1040	We can find the probability that Alice wins by finding the coefficient of x^n in the generating function.
1041	However, we can simplify this problem further by noticing that the game is essentially a sequence of independent coin flips.
1042	Let's define the probability that Alice wins as P . We can write the following equation:
1043	$P = 1/2 + (1/2)(1/2)P$
1044	Simplifying the equation, we get:
1045	$P = 1/2 + (1/4)P$
1046	[...]
1047	So, the probability that Alice wins the game is $2/3$.
1048	2
1049	3

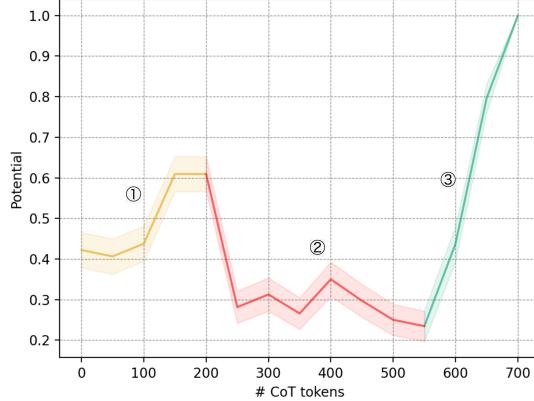


Figure 12: Reasoning tangents and insights for Llama3.1-8B on Math-500. We show an example for another dataset using Llama3.1-8B. Similar observations hold; the models tend to go on reasoning tangents as shown here with the model exploring a more difficult recursive approach, which it ultimately abandons by deriving the key insight, expressing the probability as a simple linear equation.

1061	Question	MATH-500 (Problem 42)
1062		Find the unique odd integer t such that $0 < t < 23$ and $t + 2$ is the inverse of t modulo 23.
1063	Response	
1064		To find the unique odd integer t that satisfies the given conditions, we will follow these steps: ①
1065	[...]	
1066	Add 2 to both sides: $(t + 1)^2 \equiv 2 \pmod{23}$. Since we are working modulo 23, we need to find the square roots of 2 modulo 23. We can use the fact that $2 \equiv 21 \pmod{23}$ and $21 \equiv 21^2 \pmod{23}$. We need to find x such that $x^2 \equiv 21 \pmod{23}$. By inspection, we can see that $x \equiv 5$ or $x \equiv 18$.	The model correctly finds the inverses in one shot, spiking potential ②
1067	[...]	
1068		
1069	Check if t is odd. Both $t \equiv 4$ and $t \equiv 17$ are not odd integers. ③	The model wrongly concludes that 5 is an even number
1070	[...]	
1071	Since we need $t + 2$ to be odd, we choose $t \equiv 17 \pmod{23}$. ④	The model wrongly concludes that 5 is an even number
1072	The final answer is: 17	

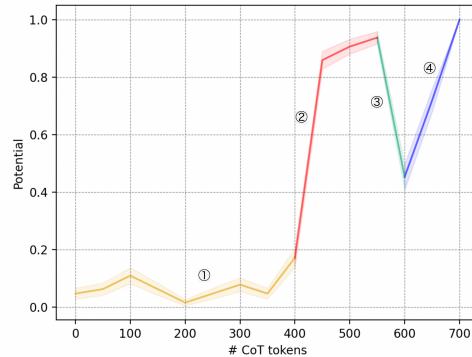


Figure 13: Reasoning tangents and insights for Llama3.1-8B on Math-500. We show an example for another dataset using Llama3.1-8B. Again the model displays a key insight, realizing directly that 5 and 18 are the square roots. A small tangent is also encountered, with the model missing that 17 is an odd number, but subsequently corrected later in the reasoning.

```

1080
1081
1082 Question (HumanEval, 151)
1083 Think step by step and complete the function corresponding to the following docstring such that it passes
1084 the given tests:
1085
1086 def double_the_difference(lst):
1087     """
1088     Given a list of numbers, return the sum of squares of the numbers
1089     in the list that are odd. Ignore numbers that are negative or not integers.
1090
1091     double_the_difference([1, 3, 2, 0]) == 1 + 9 + 0 + 0 == 10
1092     double_the_difference([-1, -2, 0]) == 0
1093     double_the_difference([9, -21]) == 81
1094     double_the_difference([0]) == 0
1095
1096     If the input list is empty, return 0.
1097     """
1098
1099 Response
1100 [...]
1101
1102 def double_the_difference(lst): ①
1103     # Filter out non-integer elements and convert them to positive integers ②
1104     filtered = [abs(int(num)) for num in lst if isinstance(num, int) and num >= 0]
1105     ③ model recovers by excluding negative ones ->
1106     # Calculate the sum of squares of odd numbers
1107     result = sum(x**2 for x in filtered if x % 2 != 0) ④
1108
1109     return result
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133

```

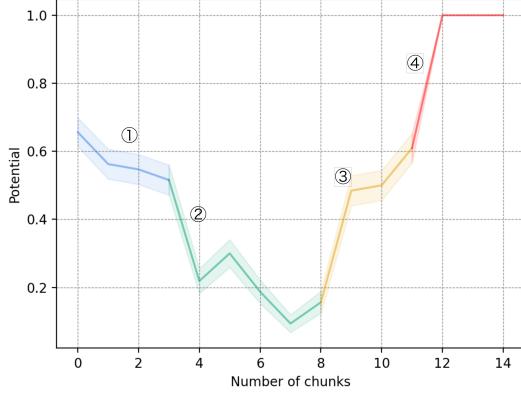


Figure 14: **Potential for coding tasks.** We analyze a trace of Qwen2.5-1.5-Instruct on HumanEval and identify the occurrence of tangents and insights. Here, the model gets distracted initially by outlining that it will convert all numbers to positive integers, and indeed using commands `abs()` and `int()`, dropping the potential significantly as a consequence. The model then recovers by fixing the for-loop using `if` statements, which actually render `abs()` and `int()` effectless as the loop anyways only goes over positive integers.

```

1109 Question (HumanEval, 128)
1110 Think step by step and complete the function corresponding to the following docstring such that it passes
1111 the given tests:
1112
1113 def prod_signs(arr):
1114     """
1115     You are given an array arr of integers and you need to return
1116     sum of magnitudes of integers multiplied by product of all signs
1117     of each number in the array, represented by 1, -1 or 0.
1118     Note: return None for empty arr.
1119
1120     Example:
1121     >>> prod_signs([1, 2, 2, -4]) == -9
1122     >>> prod_signs([0, 1]) == 0
1123     >>> prod_signs([]) == None
1124
1125 Response
1126 [...]
1127
1128 import numpy as np
1129
1130 def prod_signs(arr):
1131     if not arr:
1132         return None
1133
1134     # Calculate the product of all elements in the array ①
1135     product_of_elements = np.prod(arr)
1136
1137     # Determine the sign based on the product ②
1138     sign = 1 if product_of_elements > 0 else (-1 if product_of_elements < 0 else 0)
1139
1140     # Calculate the sum of magnitudes multiplied by the determined sign
1141     result = np.sum(np.abs(arr)) * sign
1142
1143     return result
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3
```

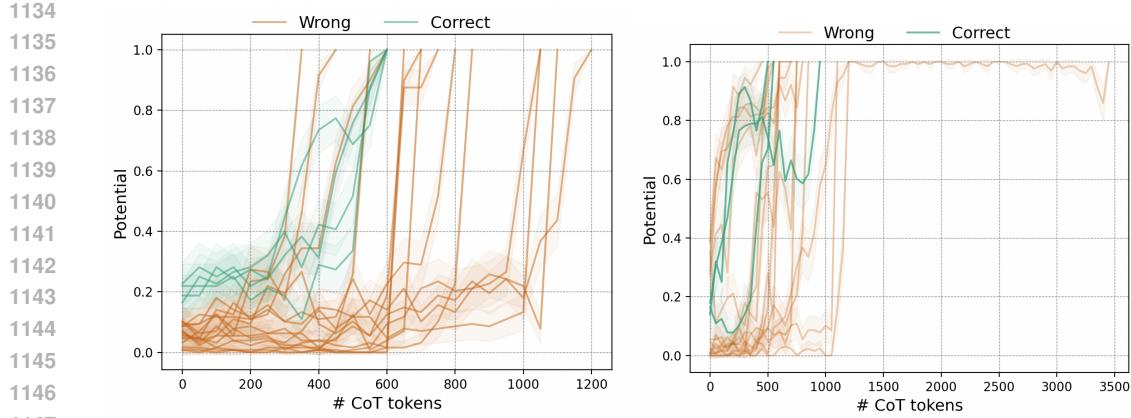


Figure 16: **Stability profiles.** Stability profiles for Qwen2.5-1.5B and Qwen2.5-7B on AIME 7 and 26 respectively. Correct and wrong answers exhibit similar profiles across models and questions.

A.4 STABILITY OF CoT

We can also consider a slight variation of the potential, called the stability of a CoT. Given a prompt x , CoT reasoning and answer pair (c, y) we define the stability of a sub-chain $c_{<t}$ as

$$\text{stable}_N(c_{<t}; x, y) := \frac{1}{N} \sum_{n=1}^N \mathbb{1}_{\{y^{(n)}=y\}} \quad \text{where } (y^{(n)}, c_{\geq t}^{(n)}) \sim \text{LM}_\theta(\cdot | c_{<t}, x)$$

with the slight variation that instead of considering the ground truth y^* , we now consider the reached final answer of the chain c as the target. I.e. the potential is a special of stability, when $y = y^*$. Stability measures how *determined* the final answer is throughout the reasoning process of the model. Somewhat surprisingly, we observe that correct answers do not necessarily always display higher stability, indicating that models can become convinced very early on in their reasoning about wrong answers. We display various stability curves in Fig. 16.

A.5 COMPUTATIONAL COMPLEXITY OF POTENTIAL CALCULATION

The computational load to obtain the potential curve of a given prompt scales with the number of evaluations N produced to estimate the potential and the number of points N_{chunks} we calculate the potential for. More precisely, say for a generation from scratch we produce T tokens, the total amount of tokens produced to estimate the potential curve will be given by

$$T_{\text{tot}} = N \sum_{i=1}^{N_{\text{chunks}}} \frac{i}{N_{\text{chunks}}} T \approx \frac{NN_{\text{chunks}}T}{2}$$

N_{chunks} hence also gives us a way to control the amount of compute and we set it to moderate values of $N_{\text{chunks}} = 15$ for the quantitative evaluations, while we use higher values of around $N_{\text{chunks}} = 50$ for most figures, to obtain a more fine-grained understanding. Advanced inference techniques such as prefix-caching and continuous batching in vLLM further help to speed up the potential calculation, as all of the prefill is shared among the prompts, and every prompt can in theory be generated in parallel, thus allowing for continuous batching. This is different for the optimal CoT, where inference has to be done sequentially as things depend on each other, making it significantly slower.

A.6 CALCULATION OF OPTIMAL CoT

We detail the recipe to calculate the optimal CoT in Algorithm 1 below:

1188 **Algorithm 1** Generating potential-optimized CoTs

1189 1: Initialize the CoT $c_{<t} \leftarrow \emptyset$

1190 2: **while** the chosen candidate does not contain the answer **do**

1191 3: Sample M candidate CoT chunks $c_{t:(t+T)}^{(m)} \stackrel{i.i.d.}{\sim} \text{LM}_\theta(\cdot \mid c_{<t}, \mathbf{x})$ of length C , for $m =$

1192 $1, \dots, M$

1193 4: Compute potentials $p_m \leftarrow \text{pot}_N(c_{<t+T}^{(m)}; \mathbf{x})$

1194 5: Select $\tilde{m} \leftarrow \arg \max_m p_m$

1195 6: Update $c_{<t+T} \leftarrow [c_{<t}, c_{t:(t+T)}^{(\tilde{m})}]$

1196 7: **end while**

1197
1198
1199 The computational complexity of the optimal CoT calculation is quite high with $\approx \frac{M N N_{\text{chunks}} T}{2}$ tokens needed, while also not enjoying the same benefits from prefix caching and continuous batching, as the potential curve calculation did. We thus do not recommend its usage in practical setting but view it more as a proof of concept.

1200 A consequence of our greedy strategy is the following: The optimal CoT could degenerate to a
1201 "lucky guessing" answer when M goes to infinity, given that the model has non-zero support across
1202 all tokens, as this would effectively minimize the objective. Since we are restricted to using a finite
1203 M however in practice, we are implicitly regularizing the objective to sequences of sufficient proba-
1204 bility mass, naturally circumventing this degeneracy. To make this regularization more explicit, one
1205 could potentially enhance the objective with a regularization term to require a given chain to meet
1206 a certain probability budget, eliminating this possibility. We observe however that M being finite is
1207 enough to avoid such degenerate cases.

1208 We do however want to stress that the optimal CoT does depend on M .

1212
1213 **A.7 MORE DETAILS ON SUMMARY STATISTICS**

1214
1215 Here we provide the definitions for the statistics we used in Sec. 4. In all experiments we divide the
1216 CoT into 20 chunks, getting thus potential curves consisting of 20 points.

- 1217
1218 • **Insight:** We say that a given potential contains an insight if the difference between two
1219 consecutive chunks of CoT exceeds 40%, i.e. if one step of CoT raised the potential by at
1220 least 40%. We exclude the last two steps to make sure we don't count the late reasoning
1221 spikes as insights.
- 1222 • **Tangent:** We define a potential to exhibit a tangent if the potential drops by at least 30%,
1223 not necessarily consecutively.
- 1224 • **Guess:** We define late reasoning spikes or guesses as the case when the potential at the
1225 second to last step is smaller than 5%.
- 1226 • **Monotonicity:** We call a potential monotone if its consecutive steps do not decrease by
1227 more than 10%.

1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241