How Well do LLMs Compress Their Own Chain-of-Thought? A Token Complexity Approach

Ayeong Lee [*1] **Ethan Che** [*1] **Tianyi Peng** [1]

Abstract

Chain-of-thought prompting has emerged as a powerful technique for enabling large language models (LLMs) to solve complex reasoning tasks. However, these reasoning chains can be verbose, raising concerns about efficiency. In this work, we conduct the first systematic study of the relationship between reasoning length and model performance across a diverse range of compression instructions. We discover a universal tradeoff between reasoning length and accuracy that persists across even very distinct reasoning chains. We demonstrate that this tradeoff emerges from a sharp threshold behavior at the question level: each task has an intrinsic 'token complexity' – a minimal number of tokens required for successful problem-solving. We use token complexity to compute upper bounds on the optimal accuracy-compression tradeoff. Our analysis reveals that prompt-based compression strategies operate far from these theoretical limits, suggesting significant room for improvement and providing benchmarks to help researchers evaluate progress in reasoning efficiency.

1. Introduction

Recent advancements in large language models (LLMs)—including o1 and DeepSeek R1—alongside broader AI agent development, have showcased impressive reasoning capabilities, hinting at a future where complex problem-solving and decision-making can be automated. However, this rapid progress also introduces a significant challenge: the computational cost of reasoning is projected to increase substantially as these models are deployed in real-world applications. This growing inference cost highlights the necessity for efficient reasoning strategies, motivating our research into reducing the computational burden of LLM inference while maintaining high performance.

A pivotal technique for enhancing LLM reasoning has been chain-of-thought (CoT) prompting, which encourages models to generate intermediate reasoning steps before arriving at a final answer (Wei et al., 2022; Kojima et al., 2022). While effective, these reasoning chains often involve lengthy intermediate computations, increasing inference costs when deployed at scale (Yu et al., 2024). Prior work has observed that asking the LLM to 'be concise' or 'use at most 100 words' can reduce response length while incurring a range of degradation to accuracy (Jin et al., 2024; Renze & Guven, 2024; Han et al., 2024). It remains unclear how best to compress the chain-of-thought content for effective problem-solving. There has not been yet a comprehensive exploration of the effectiveness of different compression strategies.

The main contribution of our work is to provide the first systematic study of the trade-off between reasoning length and performance across different prompt-based compression strategies, including prior strategies such as 'be concise' as well as alternative approaches such as 'only use bullet points' or 'use at most 50 Chinese characters'. In total, we evaluate 31 prompts for six LLMs on three standard reasoning datasets. Remarkably, although these prompting strategies produce different chains of thought, their trade-offs between response length and accuracy lie on a universal trade-off curve. In other words, all prompts are equally "good" as extremes on this curve. What primarily affects accuracy is the *length* of the chain of thought, far more than changes in its composition.

Our second contribution is a novel empirical observation: the performance of reasoning tasks exhibits a sharp threshold dependence on reasoning length at the *question* level. By evaluating multiple prompts for each question, we demonstrate that most questions have a well-defined 'token complexity'—a minimum number of tokens required to successfully solve the question–which holds across diverse prompting strategies. We estimate token complexities across various benchmarks and find that: (i) Token complexity alone can predict the performance of CoT prompting strategies

[*]Equal contribution [1]Columbia Business School, New York, USA. Correspondence to: Ayeong Lee <al3393@columbia.edu>, Ethan Che <ewc2119@columbia.edu>.

Work presented at the ES-FoMo III Workshop at ICML 2025, Vancouver, Canada. Copyright 2025 by the author(s).

Table 1: The Chain-of-Thought prompts we consider. The right column gives an example of the chain-of-thought of Claude 3.5 Sonnet on a sample problem in MMLU-Pro Math. We indicate the range of k that we consider.

PROMPT ID	PROMPT	SAMPLE OUTPUT UNDER CLAUDE 3.5 SONNET
NoCoT	ONLY GIVE THE FINAL ANSWER.	*Answer: (G)*
DefaultCoT	THINK STEP-BY-STEP.	*Let me solve this step by step: 1) Z18 (...)*
BeConcise	BE CONCISE.	*Let's solve this step by step: 1) Z18 (...)*
BulletPoints	ONLY USE BULLET POINTS.	*- Z18 is the group of integers modulo (...)*
OnlyNumbers	ONLY USE NUMBERS OR EQUATIONS.	*—Z18— = 18. Answer: (G)*
NoSpaces	DO NOT USE ANY SPACES OR LINE BREAKS.	*TheorderofagroupZnisthesame (...)*
NoProperGrammar	DO NOT USE PROPER GRAMMAR.	*lemme help u with this z18 is group*
AbbreviateWords	ABBREVIATE WORDS AS MUCH AS POSSIBLE.	*Slvng fr rdr f Z18: Z18 = grp f ntgrs (...)*
WordLimit(k)	USE AT MOST k WORDS. ($k \in [1, 100]$)	*Order of Z18 is eighteen. Answer: (G)*
CharLimit(k)	USE AT MOST k LETTERS. ($k \in [1, 500]$)	*The order of a group is the number of (...)*
TokenLimit(k)	USE AT MOST k TOKENS. ($k \in [1, 500]$)	*The order of Z18 is 18, as it contains (...)*
StepLimit(k)	USE AT MOST k STEPS. ($k \in [1, 5]$)	*Step 1: The order of a group Zn is (...)*
ChineseCoT	RESPOND IN CHINESE	让我帮你解答这个问题 (...)
ChineseCoT(k)	USE AT MOST k CHINESE CHARACTERS.	答案是18. ANSWER: (G).

with 94% accuracy. (ii) It serves as a robust measure of reasoning task difficulty, enabling us to investigate whether LLMs reason *adaptively*—using shorter chains-of-thought for easier questions.

These results raise the question of whether the trade-off curve between response-length and accuracy induced by these prompting strategies are close or far from optimal. Viewing these strategies as a form of 'lossy compression', we take inspiration from rate-distortion theory to characterize an upper bound on the optimal accuracy-compression trade-off. In doing so, we find that prompt-based strategies are far from this upper bound, especially on harder datasets.

1.1. Related Work

Recent research has begun to gain traction in exploring the trade-off between response length and accuracy in LLMs. Studies such as (Renze & Guven, 2024; Jin et al., 2024; Nayab et al., 2024; Han et al., 2024) have employed specific prompting strategies to constrain response length and assess the associated impact on accuracy and performance. Additionally, several works have highlighted the redundancy inherent in CoT prompting (Chiang & Lee, 2024; Wu et al., 2025) and emphasized the benefits of concise reasoning. Other approaches have focused on fine-tuning strategies to adapt LLMs for generating more succinct reasoning (Kang et al., 2024; Yu et al., 2024).

Our work advances this growing body of literature by making three key contributions: (1) we conducted a systematic evaluation of a rich set of prompts designed to reduce the length of CoT reasoning while maintaining accuracy. (2) We find that accurate answers are only achieved when the output length exceeds a certain threshold, which is intrinsic to the problem and independent of the CoT format. We formalize this concept as the *token complexity* of a problem. (3) We derive theoretical limits on the length-accuracy trade-off, providing a framework for researchers to benchmark new methodologies aimed at compressing chain-of-thought reasoning effectively.

2. Experiments

Our evaluation encompasses the following LLMs: GPT-4o (Hurst et al., 2024), GPT-4o-mini (Hurst et al., 2024), Claude 3.5 Sonnet (Anthropic, 2024), Claude 3.5 Haiku (Anthropic, 2024), and Llama 3.3 70B Instruct (Dubey et al., 2024). We evaluate these models on three standard math reasoning datasets: MATH-500 (Lightman et al., 2023), a random 500 problem subset of GSM8K (GSM8K, (Cobbe et al., 2021)), and a random 500 problem subset of MMLU-Pro Math problems (MMLU-Pro Math, (Wang et al., 2024)).

For each LLM and dataset, we test 31 prompts designed to induce shorter response lengths, detailed in Table 1. These prompts include ones considered in prior literature: 'be concise' (Renze & Guven, 2024), 'use k words or less' (Jin et al., 2024; Nayab et al., 2024), 'use k tokens or less' (Han et al., 2024), but include additional curated ones to assess the impact of alternative compression strategies. For each prompt, we assess performance with two metrics: (1) **accuracy**, the fraction of questions solved correctly, and (2) **average token length**, the average number of output tokens produced by the LLM in their response across questions in the dataset.

2.1. Universal Trade-off between Reasoning Length and Accuracy

By considering multiple diverse prompts, we are able to induce chains-of-thought along a spectrum of response lengths and reasoning performance, with NoCoT (no chain-of-thought) using the fewest tokens with the lowest accuracy

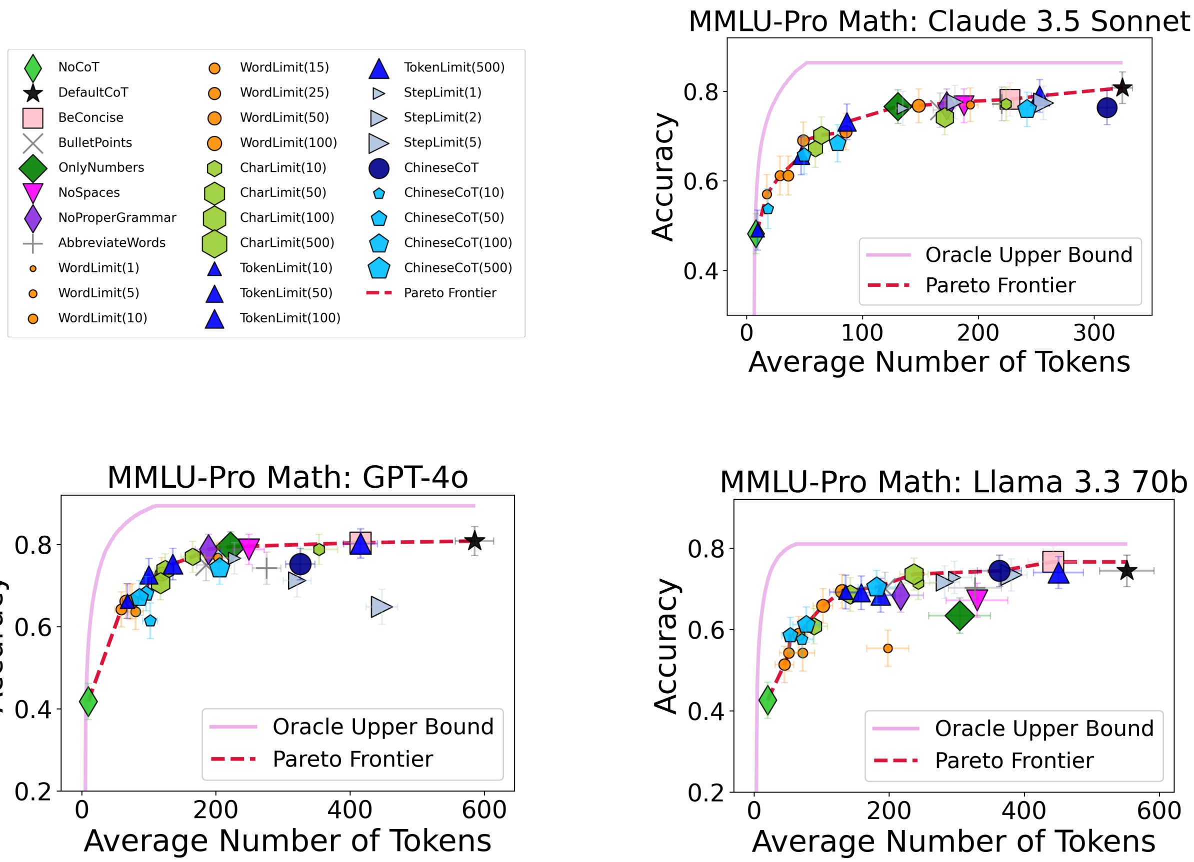

Figure 1: For each of the 31 CoT prompts we consider (see legend above), we report average token length vs accuracy for GPT-4o and Claude 3.5 Sonnet on MMLU-Pro Math and GPT-4o-mini on GSM8K tasks. Despite differences in the chain-of-thought, many of them live on a universal tradeoff curve. Using our framework, we compute upper bounds on optimal accuracy under a given average token budget (see Section 4). See Appendix G for more models and benchmarks.

and DefaultCoT (i.e. standard chain-of-though-prompting 'think step-by-step') using the most tokens and generally having the highest benchmark performance.

We observe that there is potential to achieve significant length reduction (e.g., up to 60%) compared to DefaultCoT without sacrificing much accuracy. For example, BeConcise (Renze & Guven, 2024) indeed consistently reduces token length without significantly hurting performance. The observation that the token length of DefaultCoT can be substantially improved upon without much degradation to accuracy motivates a natural question: which prompts exhibit the best tradeoff between response length and accuracy?

To study this question, we plot the average token-length and accuracy of all 31 prompts in Figure 1 for the MMLU-Pro Math benchmark (results for other benchmarks are in Appendix G). Remarkably, we see that almost all the prompts we consider lie on a *universal* trade-off curve between response length and accuracy. This suggests that regardless of whether the chain-of-thought is formatted in bullet points, without spaces, using only numbers, or even in Chinese, ultimately it is the length of the chain-of-thought that matters most. This result also holds for the Wordlimit(k), Charlimit(k), TokenLimit(k), StepLimit(k), and ChineseCoT(k) prompts, which ask the LLM to limit the response to be at most k words, letters, tokens, reasoning steps, or Chinese characters.

This universal trade-off curve suggests that the length of the chain-of-thought is the predominant factor that influences reasoning performance. We caveat this observation acknowledging that this universal trade-off should only hold for reasonably informative chains-of-thought, i.e. we would expect that pure white-space would perform worse. We also see that adherence to the universal trade-off curve is better for more capable models (i.e. GPT-4o and Claude-3.5-Sonnet) on easier benchmarks (i.e. GSM8K). For less capable models such as LLaMA 3.3 70B on harder datasets (e.g. MATH-500), there are more prompts which are below

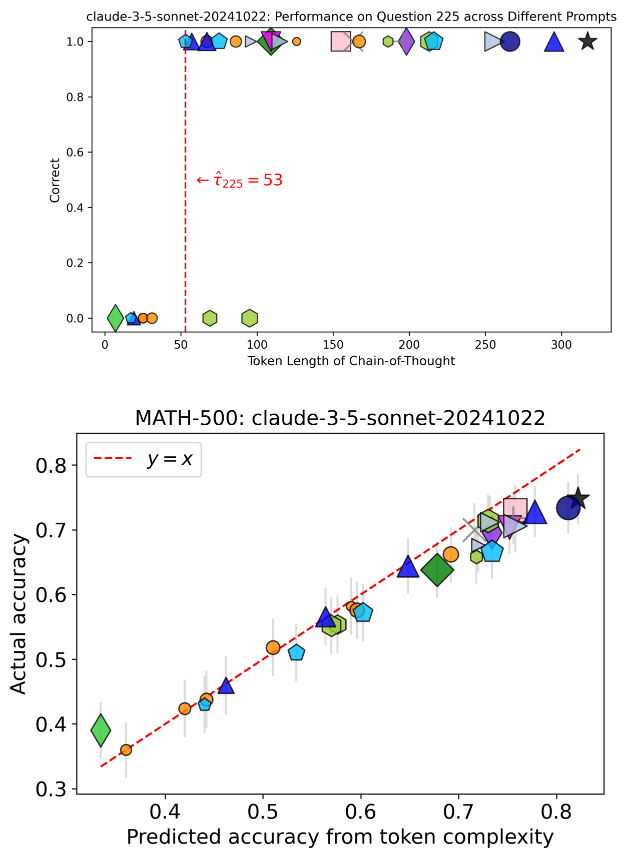

Figure 2: Illustration of the token complexity hypothesis. (Top) Performance of Claude 3.5 Sonnet on a sample question in MATH-500 exhibits a threshold behavior. The red dotted line indicates the estimated token complexity τ_i^π from the results. (Bottom) Actual benchmark accuracy on MATH-500 vs predicted accuracy from the token complexity hypothesis. Token complexity is highly predictive of actual accuracy (see Table 2).

the trade-off curve.

3. The Token Complexity Hypothesis

At the question level, we can observe that reasoning performance exhibits a sharp threshold-like behavior: across all prompts, the LLM correctly solves the question if and only if the response length is above a certain threshold. We refer to this threshold as the *token complexity* of the problem.

To illustrate, the left panel of Figure 2 displays the performance of all 31 prompts for Claude 3.5 Sonnet on a sample question in the MATH-500 dataset. Despite the diversity of prompting strategies we consider, we see that response length is highly predictive of correctness: with the exception of 2 prompts, all the prompts which use more than ≈ 53 output tokens correctly solve the problem, while the prompts that use fewer tokens get the question wrong.

To formalize this behavior, we first introduce some notation. Given a dataset of $i = 1, ..., n$ questions ($n = 500$ for the benchmarks we consider), let P_k denote a chain-of-thought prompt and let $X_{i,k}$ denote a chain-of-thought produced by an LLM π for question i when prompted by P_k. We let $t(X_{i,k}) \in \mathbb{N}$ be the length of $X_{i,k}$ in tokens. We let $a_i^\pi(X_{i,k}) = 1$ if π gets the answer correct under X_i and 0 if not. We now turn to formally outlining the 'token complexity hypothesis'.

Assumption 3.1. (Token complexity hypothesis) For question i and LLM π, there exists a threshold $\tau_i^\pi \in \mathbb{N}$, denoted as the **token complexity**, such that for any prompt P_k, the LLM gets the answer correct iff the token length $t(X_{i,k})$ is above τ_i^π:

$$a_i^\pi(X_{i,k}) = \mathbf{1}\{t(X_{i,k}) \geq \tau_i^\pi\}$$

We proceed to test this hypothesis quantitatively across LLMs π and benchmarks. First, we measure to what degree success or failure on a task can be classified based purely on the token-length of the chain-of-thought, i.e. whether the behavior in the left panel of Figure 2 holds broadly. To do so, we use our dataset of $k = 1, ..., K = 31$ chain-of-thought prompts for each question. For a threshold $t \in \mathbb{N}$, we define $c_i^\pi(t)$ to be the classification accuracy under a threshold classifier under t, which measures how predictable reasoning success is based on whether the token count exceeds threshold t

$$c_i^\pi(t) \equiv \frac{1}{K}\sum_{k=1}^{K} \mathbf{1}\{a_i^\pi(X_{i,k}) = \mathbf{1}\{t(X_{i,k}) \geq t\}\}$$

Our estimator $\hat{\tau}_i^\pi$ of token complexity is the optimal threshold-based classifier, and we let c_i^* be the maximum classification accuracy achieved by $\hat{\tau}_i^\pi$.

$$\begin{aligned} \hat{\tau}_i^\pi &\equiv \arg\max_k c_i^\pi(t(X_{i,k})), \qquad (1) \\ c_i^* &\equiv \max_k c_i^\pi(t(X_{i,k})), \\ \bar{c}_\pi &\equiv \frac{1}{n}\sum_{i=1}^{n} c_i^* \end{aligned}$$

We let $\hat{\tau}_i^\pi = \infty$ if setting the threshold to $t = \infty$ results in better classification accuracy, e.g. if none of the chains-of-thoughts correctly solve the problem in which case $a_i^\pi(X_{i,k}) = 0$ for all k. For each dataset and model, we report the average classification accuracy $\bar{c}_\pi$ under the estimated τ_i^π in Table 3. Overall, the average classification accuracy is very high, above 90% for most models and benchmarks, verifying that (1) token-length is highly predictive for performance at the question-level and (2) question-level performance exhibits a threshold relationship with token-length. We also see that the threshold classifier

Table 2: Evidence of token complexity hypothesis across LLMs and reasoning benchmarks. The high classification accuracy of $\bar{c}_\pi \approx 90\%$ and the small discrepancy $\text{Err}_\pi \approx 6\%$ illustrate that the token complexity hypothesis has strong predictive power for the reasoning performance of LLMs at both the question and benchmark levels.

Model	MMLU-Pro Math		GSM8K		MATH-500	
	$\bar{c}_\pi$	Err_π	$\bar{c}_\pi$	Err_π	$\bar{c}_\pi$	Err_π
GPT-4o	92.1%	5.0%	97.0%	1.6%	89.9%	6.0%
GPT-4o Mini	91.3%	5.2%	94.1%	5.1%	91.4%	5.7%
Claude 3.5 Sonnet	92.6%	3.4%	97.4%	1.4%	92.1%	4.5%
Claude 3.5 Haiku	90.1%	5.1%	94.8%	4.1%	90.2%	6.3%
Llama 3.3 70B	90.6%	5.1%	96.2%	2.7%	89.8%	6.1%

achieves higher average classification accuracy for (1) more capable models (i.e. GPT-4o and Claude 3.5 Sonnet) and (2) easier benchmarks (i.e. GSM8K), which mirrors the finding in Section 2.1 concerning adherence to the universal trade-off curve.

Our second empirical validation is to compare the accuracy of the LLM across prompts and benchmarks with the performance predicted under the token complexity hypothesis. We let $\text{Acc}_\pi(P_k)$ denote the accuracy of LLM π under prompt P_k while $\widehat{\text{Acc}}_\pi(P_k)$ denotes the accuracy predicted under our estimated token-complexities:

$$\text{Acc}_\pi(P_k) \equiv \frac{1}{n}\sum\nolimits_{i=1}^{n} a_i^\pi(X_{i,k}), \tag{2}$$
$$\widehat{\text{Acc}}_\pi(P_k) \equiv \frac{1}{n}\sum\nolimits_{i=1}^{n} \mathbf{1}\{t(X_{i,k}) \geq \hat{\tau}_i^\pi\}$$

In Table 2, for each LLM and dataset we report Err_π, the average relative discrepancy between actual accuracy and predicted accuracy:

$$\text{Err}_\pi = \frac{1}{K}\sum\nolimits_{k=1}^{K} \frac{|\text{Acc}_\pi(P_k) - \widehat{\text{Acc}}_\pi(P_k)|}{\text{Acc}_\pi(P_k)}$$

We observe that the token threshold classifier is able to predict overall benchmark performance within 6% error, and the error is even smaller for larger models on easier benchmarks. This illustrates that the token-complexity hypothesis provides an accurate model for reasoning performance, which can be seen visually in the right panel of Figure 2 for Claude 3.5 Sonnet on the MATH-500 dataset and helps characterize the strong dependence of token-length on accuracy. We explore the implications of this hypothesis in the next section.

4. Theoretical Limits of the Length-Performance Tradeoff

Not only does token-complexity provide an interpretable and accurate model of reasoning task performance, it gives insight into how to improve efficiency. We develop a framework to empirically compute bounds on compression performance, inspired by rate-distortion theory. First, we define $\bar{t}_\pi(P) \equiv \frac{1}{n}\sum_{i=1}^n t(X_i)$ to be the average token-length under CoT prompt P and LLM π. We define $\alpha^*(T)$ to be the optimal performance for an *average* token budget of T tokens, and $T^*(\alpha)$ is the minimum average token length to achieve an accuracy of α:

$$\alpha_\pi^*(T) = \max_P\{\text{Acc}_\pi(P) : \bar{t}_\pi(P) \leq T\} \tag{3}$$
$$T_\pi^*(\alpha) = \min_P\{\bar{t}_\pi(P) : \text{Acc}_\pi(P) \geq \alpha\} \tag{4}$$

These quantities involve intractable optimizations over CoT prompts P. Yet, under the token complexity hypothesis, these optimization problems are greatly simplified, involving only optimization over token-lengths. In fact, the optimization problem is a special case of a knapsack problem, and thus gives rise to a closed-form solution. First, define the empirical CDF of the true token complexities $F_n(t) \equiv \frac{1}{n}\sum_{i=1}^n \mathbf{1}\{\tau_i^\pi \leq t\}$, let $E_n(t) \equiv \frac{1}{n}\sum_{i=1}^n \tau_i^\pi \mathbf{1}\{\tau_i^\pi \leq t\}$, and let $\bar{\tau}_\pi = \frac{1}{N}\sum_{i=1}^n \tau_i^\pi \mathbf{1}\{\tau_i^\pi < \infty\}$ denote the average token complexity among questions that have finite token complexity.

Theorem 4.1. *Suppose Assumption 3.1 and suppose that for any token counts $\{t_i\}_{i=1}^n$ there exists a prompt P such that $t(X_i) = t_i$. Then,*

$$\alpha_\pi^*(T) = \frac{1}{n}\sum\nolimits_{i=1}^{n} \mathbf{1}\{\tau_i^\pi \leq t_T\} \tag{5}$$
$$T_\pi^*(\alpha) = \frac{1}{n}\sum\nolimits_{i=1}^{n} \tau_i^\pi \mathbf{1}\{\tau_i^\pi \leq q_\alpha\} \tag{6}$$

where $t_T \equiv \sup\{t \in \mathbb{R} : E_n(t) \leq T\}$ and $t_T = \infty$ if $T > \bar{\tau}^\pi$ and $q_\alpha = \sup\{t \in \mathbb{R} : F_n(t) \leq \alpha\}$ is the α-quantile of the empirical distribution.

Intuitively, under Assumption 3.1 the optimal strategy is to only use the minimal number of tokens required to solve the problem, τ_i^π. Under a limited token budget, it is optimal to sort questions by token complexity and solve the questions with shortest token complexity until the budget is filled. Note that this structure is due to the fact that each question is weighted identically, if certain questions had a higher weight than others (e.g. harder questions are more valuable) then the optimal strategy need not have a closed form solution.

Table 3: Comparison of token counts on the Math-500 dataset under DefaultCoT and BeConcise along with $T^*_\pi(A^*)$, which is a lower bound on the average token length required to achieve max accuracy. While existing prompt strategies do meaningfully reduce token-length, the lower bound $T^*_\pi(A^*)$ illustrates that one can achieve drastically more compression while preserving accuracy.

Dataset	Model	DefaultCoT Token Count	BeConcise Token Count	$T^*_\pi(A^*)$	BeConcise Token Reduction	Upper Bound of Token Reduction
Math-500	GPT-4o	635	505	172	1.26x	3.69x
	GPT-4o-mini	611	528	164	1.16x	3.72x
	Claude-3.5-Sonnet	373	283	105	1.32x	3.56x
	Claude-3.5-Haiku	373	287	143	1.30x	2.61x
	Llama-3.3-70B	549	475	93	1.16x	5.88x

We note that it is unlikely that any feasible prompting technique will achieve the upper bound $\alpha^*_\pi(T)$ or the lower bound $T^*(\alpha)$, as doing so requires (1) knowing the token complexities, (2) inducing a chain-of-thought that exactly matches the token complexity, (3) prioritizing easier questions over harder ones.

Nevertheless $\alpha^*_\pi(T)$ provides a computable upper bound on maximum accuracy for a particular token budget, which we plot in Figure 1 under the label 'oracle upper bound', using estimated token complexities $\hat{\tau}_i^\pi$. Across all the LLMs and benchmarks we consider, we find that this is indeed serves as an upper bound on the performance of the prompting strategies we consider across response lengths, especially for more difficult datasets such as MATH-500 and MMLU-Pro Math. Yet, for GSM8K in Appendix G, we see that this gap with the upper bound $\alpha^*_\pi(T)$ is much smaller, illustrating that while $\alpha^*_\pi(T)$ may be challenging to achieve it still gives a reasonable upper bound on performance.

Finally, we circle back to the observation made in Section 2.1 that one can substantially reduce the length of the chain-of-thought (with prompts like BeConcise or NoProperGrammar) while maintaining a similar accuracy to DefaultCoT. This leads to a natural question: what is the lower bound on the number of tokens needed in order to achieve the best possible accuracy? Under the token-complexity hypothesis, we obtain a simple closed-form expression for this lower bound.

Corollary 4.2. *Suppose Assumption 1 holds. Let* $A^* = \frac{1}{n}\sum_{i=1}^n \mathbf{1}\{\tau_i^\pi < \infty\}$ *be the maximum possible accuracy achieved by the LLM on the dataset. The number of tokens required to achieve accuracy* $\alpha = A^*$ *under the optimal token allocation:*

$$T^*_\pi(A^*) = \bar{\tau}_\pi = \frac{1}{n}\sum_{i=1}^n \tau_i^\pi \mathbf{1}\{\tau_i^\pi < \infty\} \quad (7)$$

Corollary 4.2 gives the number of tokens required for 'lossless compression', i.e. achieving the best possible accuracy. Remarkably, this lower bound is only the *mean* of the token-complexities, which can be much smaller than the average token-length of existing prompting strategies. In Table 3, we consider the MMLU-Pro Math dataset across several LLMs, and we compare the lower bound $T^*_\pi(A^*)$ with the token counts of DefaultCoT and BeConcise. Across all the LLMs, while BeConcise reduces the token counts by 1.2-1.4x, the token reduction achieved by the optimal compression scheme is **3.5-5.8x**. Along with the results in Figure 1, this illustrates that while the upper bound may not be exactly attainable, there may be significant room for improvement especially for easier datasets.

Finally, we note that token complexity may be of broader interest as a measure of reasoning capabilities, even for benchmarks which are 'saturated'. For instance, both Claude 3.5 Sonnet and Claude 3.5 Haiku achieve a similar accuracy on GMS8K (97% vs 95% under DefaultCoT). Yet the average token complexity $\bar{\tau}_\pi$ is 42.4 for Haiku and 17.9 for Sonnet, showing that even if the accuracy is similar, Sonnet can achieve it with much fewer tokens.

5. Towards the Theoretical Limit

The token complexity hypothesis not only provides limits on the efficiency of an optimal chain-of-thought compression scheme, it highlights the importance of *adaptive* compression – using shorter chains-of-thought for easier questions – for approaching these limits. This provides theoretical motivation for methods developed in recent works (e.g. (Han et al., 2024; Kang et al., 2024)) designed to calibrate the length of the chain-of-thought to problem difficulty. In this section, we illustrate how our empirical framework can help evaluate and contextualize recent advances in adaptive reasoning.

As a proof of concept, we consider two prompting strategies designed to adjust reasoning effort to problem difficulty. (Han et al., 2024) propose a two-step procedure called TALE-EP, which first (1) prompts the LLM to guess the minimum tokens required to solve the question and then (2) prompts the LLM to think step-by-step using the guessed number of tokens. Using our experimental results, we can compare whether adaptively adjusting the reasoning effort improves upon the accuracy-length tradeoff. The left

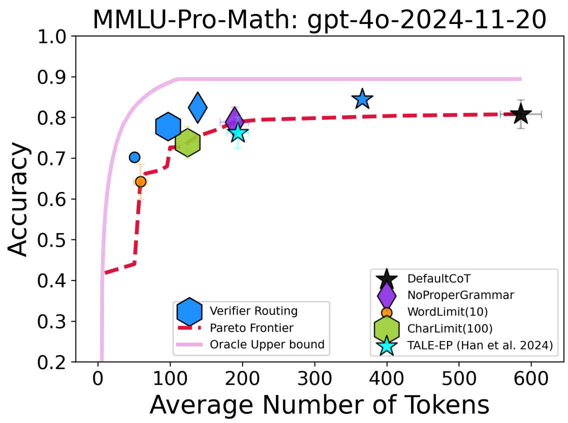

Figure 3: Does adaptive compression improve the accuracy-length tradeoff? (Top) Average token length vs Accuracy for GPT-4o on MMLU-Pro-Math under different prompt routing strategies. TALE-EP (Han et al., 2024) (light green), which first has the LLM guess the number of tokens to use, is slightly below the Pareto curve achieved by simpler prompting strategies. Only verifier-based routing (in blue) is able to achieve a better tradeoff, closer to the upper bound.

panel of Figure 3 plots accuracy and average token-length of TALE-EP for GPT-4o on MMLU-Pro Math compared to several prompts from our experiments. Surprisingly, we observe that TALE-EP is slightly below the trade-off curve from our simple prompting strategies. More specifically, it produces a slightly worse accuracy than NoProperGrammar, even though it uses more tokens on average. This is consistent across other benchmarks (see Appendix H).

We find that that the token counts produced by NoProperGrammar have a surprisingly high correlation with the true token complexity, which is equivalent to the correlation for the token counts of TALE-EP (Spearman $\rho = 0.51$ for TALE-EP and $\rho = 0.54$ for NoProperGrammar, see Appendix B for other prompts). Thus, this shows that LLMs *natively* adjust response length to the difficulty of the problem, even without explicitly being prompted to do so. And moreover, this capability is *equivalent* to more sophisticated prompting strategies, demonstrating that the simple prompting strategies we test are surprisingly strong baselines and that LLMs may struggle to estimate token complexity accurately. Nonetheless, methods based on fine-tuning (e.g. (Kang et al., 2024) or TALE-PT in (Han et al., 2024)) may be able to outperform these simple prompting strategies, which we leave to future research.

This result suggests that it may be challenging to achieve a better accuracy-length tradeoff. Nonetheless, if one has access to a perfect verifier (Brown et al., 2024; Setlur et al., 2024), we can substantially improve the tradeoff. Verifier Routing first (1) prompts the LLM to without chain-of-thought (NoCoT) to obtain an initial solution and (2) if the solution is incorrect it uses a longer chain-of-thought prompt (e.g. DefaultCoT) to produce a better answer. The blue dots in the left panel Figure 3 show performance of Verifier Routing with a selection of four longer prompts. This routing strategy achieves a significantly better trade-off between reasoning length and performance, and approaches the theoretical upper bound. Altogether, this shows that the upper bound $\alpha^*_\pi(T)$ can be approached through adaptive compression, although this requires a very accurate signal of problem difficulty and motivates further research.

6. Conclusion

Our study presents a systematic investigation into the trade-off between reasoning length and performance in large language models (LLMs), across different prompts. We demonstrate that this trade-off follows a universal Pareto curve, suggesting that reasoning length, rather than specific compression strategies, primarily determines accuracy. Introducing the concept of token complexity, we find that LLM performance at the question-level exhibits a sharp threshold behavior. Our analysis, inspired by rate-distortion theory, reveals that existing prompt-based compression strategies operate far from the optimal accuracy-length frontier, highlighting substantial room for improvement. Our work enables researchers to contextualize the performance of new methodologies for improving chain-of-thought compression and assess adaptivity of LLM reasoning.

Impact Statement

Our paper advances the understanding of the tradeoffs between reasoning efficiency and accuracy in large language models. By introducing the concept of token complexity and providing theoretical and empirical tools to assess reasoning compression, this work helps researchers benchmark improvements in reasoning efficiency. While we do not foresee immediate ethical risks, improving the efficiency of LLM reasoning could have downstream societal impacts such as enabling broader access to high-performance language models in resource-constrained settings.

References

Anthropic, A. Claude 3.5 sonnet model card addendum. *Claude-3.5 Model Card*, 3(6), 2024.

Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q. V., Ré, C., and Mirhoseini, A. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Chiang, C.-H. and Lee, H.-Y. Over-reasoning and redundant calculation of large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 161–169, 2024.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Han, T., Fang, C., Zhao, S., Ma, S., Chen, Z., and Wang, Z. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*, 2024.

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Jin, M., Yu, Q., Shu, D., Zhao, H., Hua, W., Meng, Y., Zhang, Y., and Du, M. The impact of reasoning step length on large language models. *arXiv preprint arXiv:2401.04925*, 2024.

Kang, Y., Sun, X., Chen, L., and Zou, W. C3ot: Generating shorter chain-of-thought without compromising effectiveness. *arXiv preprint arXiv:2412.11664*, 2024.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Nayab, S., Rossolini, G., Buttazzo, G., Manes, N., and Giacomelli, F. Concise thoughts: Impact of output length on llm reasoning and cost. *arXiv preprint arXiv:2407.19825*, 2024.

Renze, M. and Guven, E. The benefits of a concise chain of thought on problem-solving in large language models. *arXiv preprint arXiv:2401.05618*, 2024.

Setlur, A., Nagpal, C., Fisch, A., Geng, X., Eisenstein, J., Agarwal, R., Agarwal, A., Berant, J., and Kumar, A. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*, 2024.

Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Wu, Y., Wang, Y., Du, T., Jegelka, S., and Wang, Y. When more is less: Understanding chain-of-thought length in llms. *arXiv preprint arXiv:2502.07266*, 2025.

Yu, P., Xu, J., Weston, J., and Kulikov, I. Distilling system 2 into system 1. *arXiv preprint arXiv:2407.06023*, 2024.

A. Limitations

While our study provides novel insights into the relationship between reasoning length and LLM performance, there are several limitations. First, our study is limited to mathematical reasoning tasks, and it remains to be seen whether similar accuracy-length tradeoffs and token complexity thresholds apply to other domains such as commonsense reasoning, code generation, or open-ended text generation. The theoretical upper bound on accuracy-length tradeoff is rarely achieved due to the practical challenges in estimating token complexities and generating precisely compressed responses. Computing token complexity is computationally expensive, requiring multiple generations per question, making it impractical for large-scale applications. Our approach also assumes that token complexity is well-defined and consistent across different models and tasks, which may not hold for all LLMs or highly complex benchmarks. Furthermore, our experiments were limited to a fixed set of 31 prompts, and exploring a broader range of compression strategies, including model fine-tuning or iterative refinement, could potentially expand the compression frontier. Finally, our analysis is most relevant for strong models on moderately difficult benchmarks; weaker models or extremely challenging tasks may exhibit different behavior.

B. Correlation with Token Complexity

The following is a table describing the Spearman correlation between token-lengths with token-complexity across different prompts for GPT-4o on MMLU-Pro Math. While the ordering of which prompts have the highest correlation changes across different models and benchmarks, we observe widely that the correlation ranges between 0 - 0.6.

Prompt ID	Spearman ρ
CharLimit(50)	0.57
NoSpaces	0.56
OnlyNumbers	0.55
CharLimit(100)	0.55
NoProperGrammar	0.54
⋮	⋮
TokenLimit(10)	0.38
WordLimit(15)	0.37
WordLimit(5)	0.29
WordLimit(10)	0.28
NoCoT	0.08

Table 4: GPT-4o on MMLU-Pro Math: Spearman correlation of token-lengths with token-complexity across different prompts.

In Figure 4, we consider the average token lengths of GPT-4o on MMLU-Pro Math. We split the problems into two categories: problems that the LLM successfully solves without chain-of-thought (NoCot), and the problems NoCot unsuccessfully solves. Intuitively, the first class of problems are 'easy', since they do not require any chain-of-thought, and the rest are harder. Across all the prompts we consider, we observe that the average token-length among 'easy' problems is universally smaller than among 'harder' problems. Surprisingly, this is even true for the WordLimit/TokenLimit/CharLimit/etc. prompts, which are supposed to limit the LLM's response to a fixed length. Nevertheless, despite the evidence of adaptive response lengths, this also shows that there is a lot of room for improvement: even though the LLM can solve the problem without any chain-of-thought, the model still proceeds to generate long chains-of-thought.

C. Accuracy and Average Token Length

Below are tables which describe the accuracy and average token length for several prompts across models and benchmarks.

D. Proof of Theorem 1

Under Assumption 3.1, we have that the accuracy can be represented as

$$\mathrm{Acc}_\pi(P) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{\tau_i \geq t_i\} \tag{8}$$

Model	Prompt	Accuracy	Token Length
GPT-4o	NoCoT	41.8	9
	DefaultCoT	80.8	585
	BeConcise	80.4	415
	BulletPoints	75.0	185
	OnlyNumbers	79.4	221
	NoSpaces	78.8	248
	NoProperGrammar	78.8	189
	AbbreviateWords	74.2	275
GPT-4o Mini	NoCoT	32.4	6
	DefaultCoT	75.0	506
	BeConcise	74.8	418
	BulletPoints	63.0	128
	OnlyNumbers	68.8	189
	NoSpaces	66.2	219
	NoProperGrammar	61.0	119
	AbbreviateWords	56.6	259
Claude 3.5 Sonnet	NoCoT	48.2	7
	DefaultCoT	80.8	324
	BeConcise	78.2	227
	BulletPoints	75.8	167
	OnlyNumbers	76.6	130
	NoSpaces	76.8	187
	NoProperGrammar	76.8	172
	AbbreviateWords	77.2	220
Claude 3.5 Haiku	NoCoT	30.2	19
	DefaultCoT	68.8	296
	BeConcise	68.4	237
	BulletPoints	61.8	155
	OnlyNumbers	63.2	147
	NoSpaces	62.6	193
	NoProperGrammar	63.2	169
	AbbreviateWords	62.6	243
LLaMA 3.3 70B	NoCoT	42.6	20
	DefaultCoT	74.4	551
	BeConcise	76.6	442
	BulletPoints	70.0	193
	OnlyNumbers	63.4	304
	NoSpaces	67.2	330
	NoProperGrammar	68.4	217
	AbbreviateWords	70.2	327

Table 5: Accuracy-length tradeoff on MMLU-Pro Math

Model	Prompt	Accuracy	Token Length
GPT-4o	NoCoT	70.0	30
	DefaultCoT	92.8	266
	BeConcise	96.6	190
	BulletPoints	96.6	104
	OnlyNumbers	97.4	76
	NoSpaces	96.4	93
	NoProperGrammar	95.8	71
	AbbreviateWords	95.0	97
GPT-4o Mini	NoCoT	28.0	6
	DefaultCoT	94.6	292
	BeConcise	94.2	216
	BulletPoints	92.2	97
	OnlyNumbers	92.0	77
	NoSpaces	86.6	62
	NoProperGrammar	91.0	76
	AbbreviateWords	72.8	121
Claude 3.5 Sonnet	NoCoT	67.8	7
	DefaultCoT	97.0	200
	BeConcise	97.4	136
	BulletPoints	97.0	100
	OnlyNumbers	96.0	66
	NoSpaces	97.0	111
	NoProperGrammar	97.8	111
	AbbreviateWords	95.0	119
Claude 3.5 Haiku	NoCoT	30.6	8
	DefaultCoT	95.2	211
	BeConcise	94.4	169
	BulletPoints	92.8	114
	OnlyNumbers	93.0	87
	NoSpaces	92.4	116
	NoProperGrammar	91.8	120
	AbbreviateWords	90.8	137
LLaMA 3.3 70B	NoCoT	88.6	98
	DefaultCoT	96.2	194
	BeConcise	95.8	147
	BulletPoints	96.2	80
	OnlyNumbers	89.6	81
	NoSpaces	92.2	102
	NoProperGrammar	95.4	91
	AbbreviateWords	94.0	136

Table 6: Accuracy-length tradeoff on GSM8K

Model	Prompt	Accuracy	Token Length
GPT-4o	NoCoT	55.6	116
	DefaultCoT	72.8	634
	BeConcise	71.6	505
	BulletPoints	70.6	302
	OnlyNumbers	68.4	272
	NoSpaces	71.2	368
	NoProperGrammar	68.8	286
	AbbreviateWords	72.0	416
GPT-4o Mini	NoCoT	25.6	9
	DefaultCoT	70.4	610
	BeConcise	72.0	528
	BulletPoints	68.8	266
	OnlyNumbers	66.2	306
	NoSpaces	67.6	390
	NoProperGrammar	66.8	254
	AbbreviateWords	62.0	367
Claude 3.5 Sonnet	NoCoT	39.0	9
	DefaultCoT	74.8	373
	BeConcise	73.0	282
	BulletPoints	70.0	203
	OnlyNumbers	63.8	172
	NoSpaces	70.4	240
	NoProperGrammar	69.6	225
	AbbreviateWords	71.8	263
Claude 3.5 Haiku	NoCoT	25.8	19
	DefaultCoT	66.0	373
	BeConcise	64.0	286
	BulletPoints	60.0	243
	OnlyNumbers	59.0	193
	NoSpaces	56.4	257
	NoProperGrammar	61.4	253
	AbbreviateWords	59.4	294
LLaMA 3.3 70B	NoCoT	33.8	50
	DefaultCoT	55.4	549
	BeConcise	67.0	475
	BulletPoints	63.6	238
	OnlyNumbers	57.0	362
	NoSpaces	57.2	439
	NoProperGrammar	61.2	294
	AbbreviateWords	63.8	383

Table 7: Accuracy-length tradeoff on MATH-500

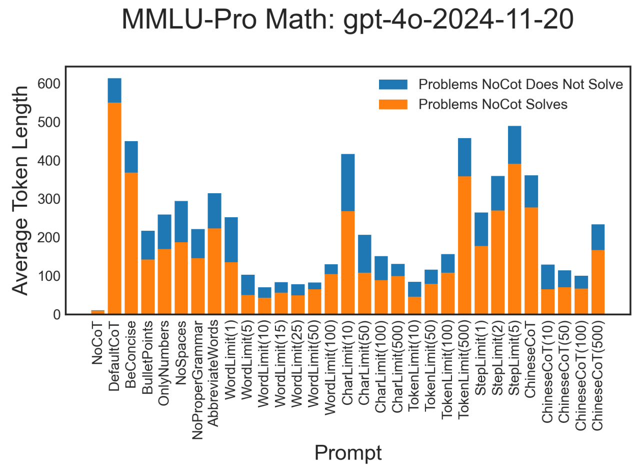

Figure 4: Do LLM's tailor response length to problem difficulty? Average token length produced by GPT-4o on MMLU-Pro Math across prompts in Table 1, split by problems which can be solved without chain-of-thought and problems which NoCoT does not successfully solve. Response lengths are consistently higher for the problems NoCoT unsuccessfully solves.

where we suppress dependence on π for now. The optimization problem then becomes

$$\alpha^*_\pi(T) = \max_t \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{\tau_i \geq t_i\} \tag{9}$$

$$\text{s.t. } \frac{1}{n}\sum_{i=1}^{n} t_i \leq T \tag{10}$$

Note that the optimal strategy is either to set $t_i = \tau_i$ or 0. Thus, this is equivalent to a Knapsack problem,

$$\alpha^*_\pi(T) = \max_{x_i \in \{0,1\}} \frac{1}{n}\sum_{i=1}^{n} x_i \tag{11}$$

$$\text{s.t. } \frac{1}{n}\sum_{i=1}^{n} \tau_i x_i \leq T \tag{12}$$

where x_i are indicators whether the LLM tries to solve the problem or not. This is a special case where there are unit rewards (since all problems are weighted equally). The optimal strategy in this case is a greedy policy, sorting the questions in increasing order of token complexity $\tau_{(1)} < ... < \tau_{(n)}$ and only solving as many as can fit within budget. $t_T \equiv \sup\{t \in \mathbb{R} : E_n(t) \leq T\}$ is the highest value of τ that the LLM puts into the knapsack after ordering them greedily, so thus $\alpha^*_n(T)$ is the total number of questions with token complexity less than τ. The proof for $T^*(\alpha)$ proceeds similarly, as the optimal strategy is identical.

E. Example Prompt

We use the following template for our prompts:

> Answer the following question. PROMPT Question: QUESTION The last line of your response should be of the following format: 'Answer: ANSWER' (without quotes) where ANSWER is your final answer.

F. Tradeoff curves for more models and benchmarks

In this section, we present results for the performance of different models and benchmarks (GSM8K and MATH-500). We see broadly that performance across all prompts lies on the same trade-off curve.

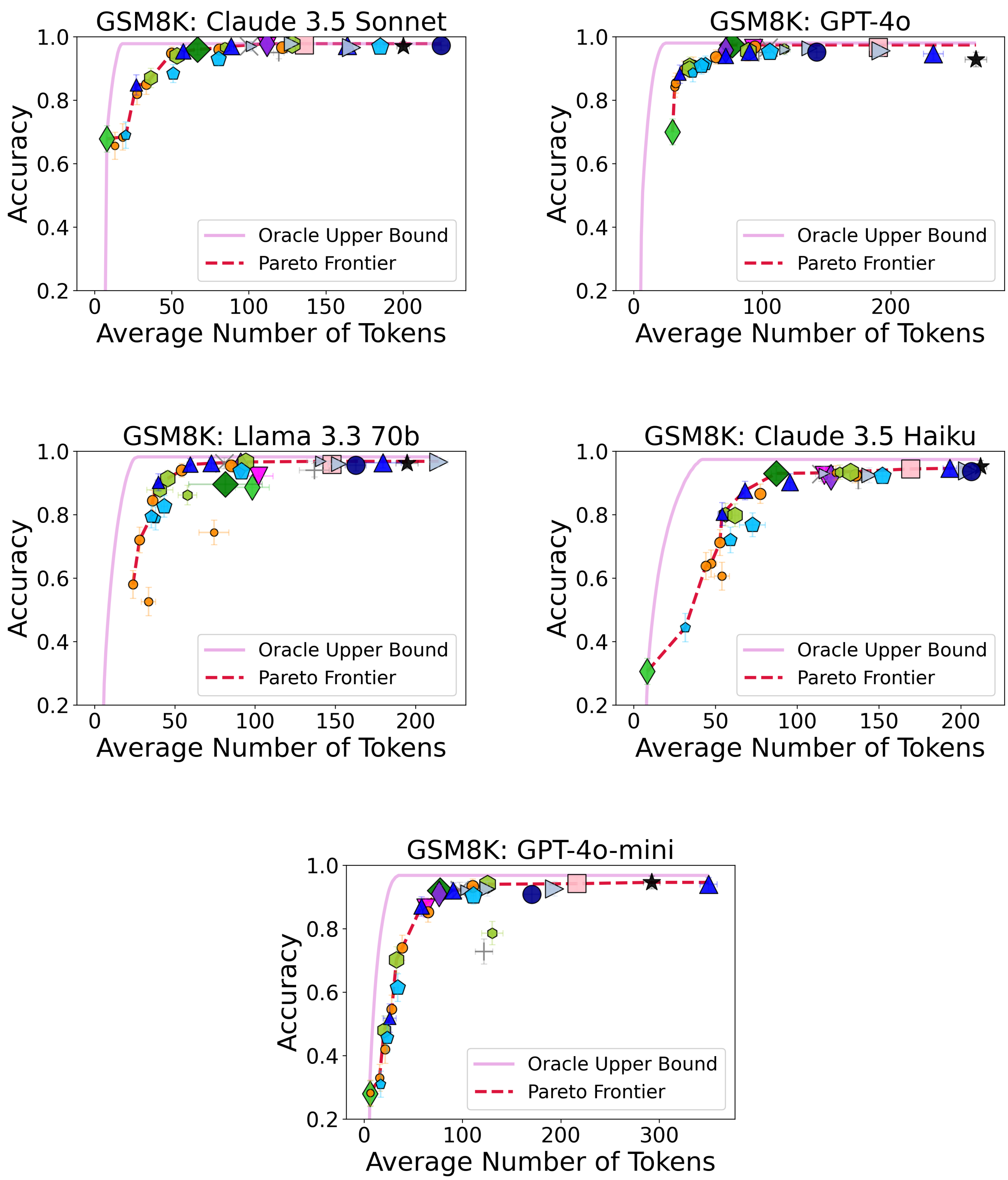

Figure 5: Tradeoff Curves for GSM8K

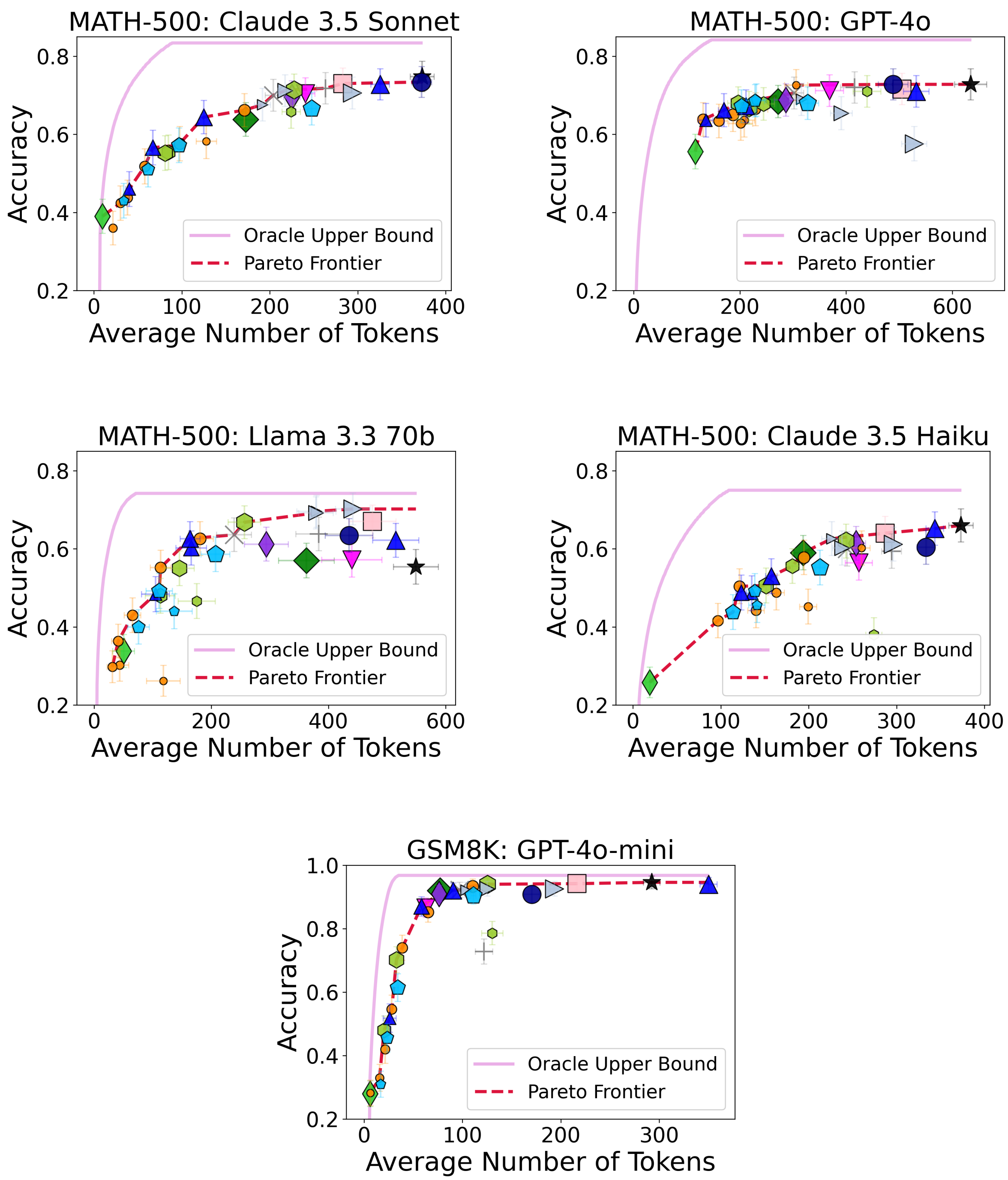

Figure 6: Tradeoff Curves for MATH-500

G. Actual accuracy vs Predicted Accuracy from Token Complexity

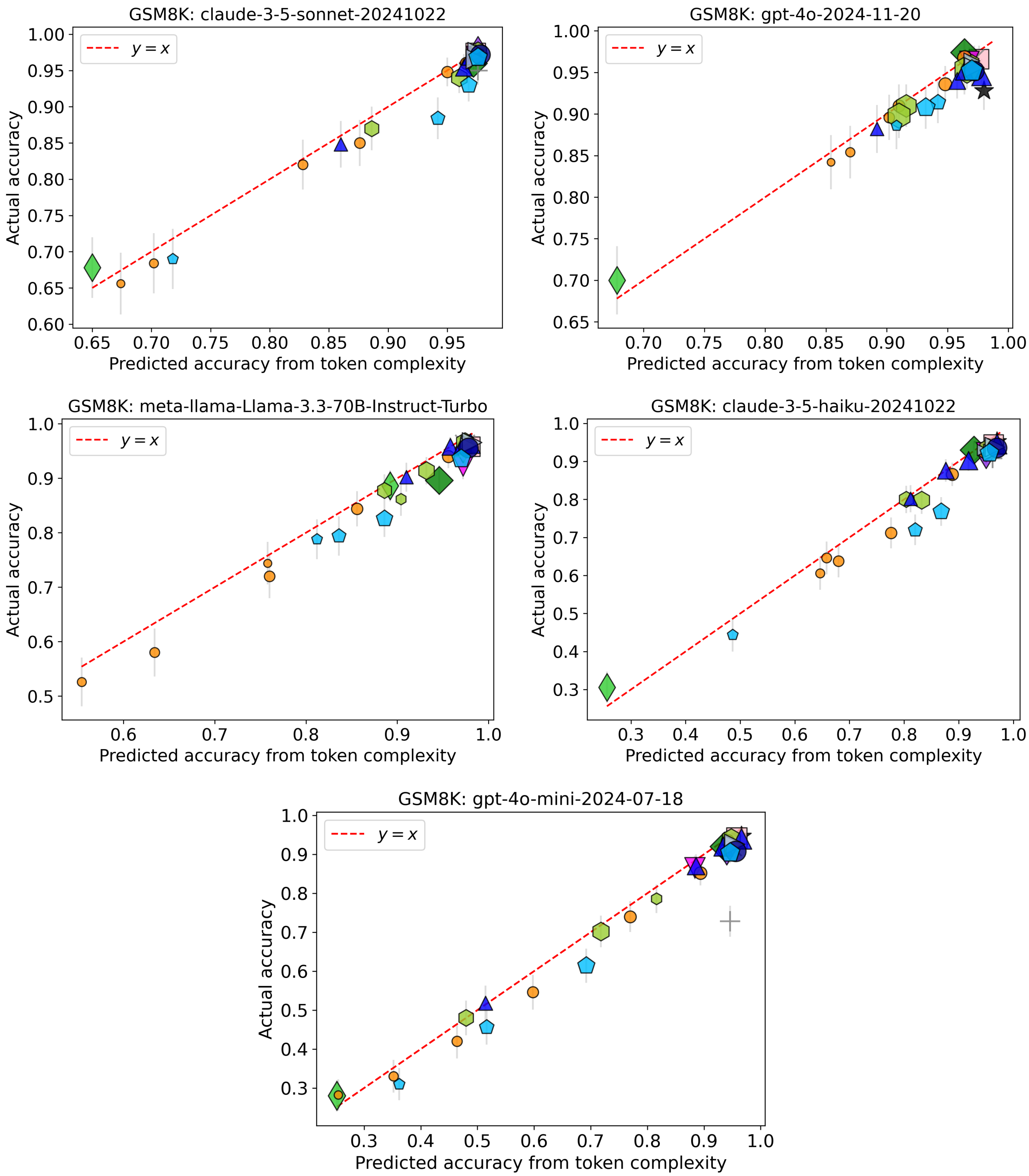

Figure 7: Actual vs Predicted Accuracy for GSM8k

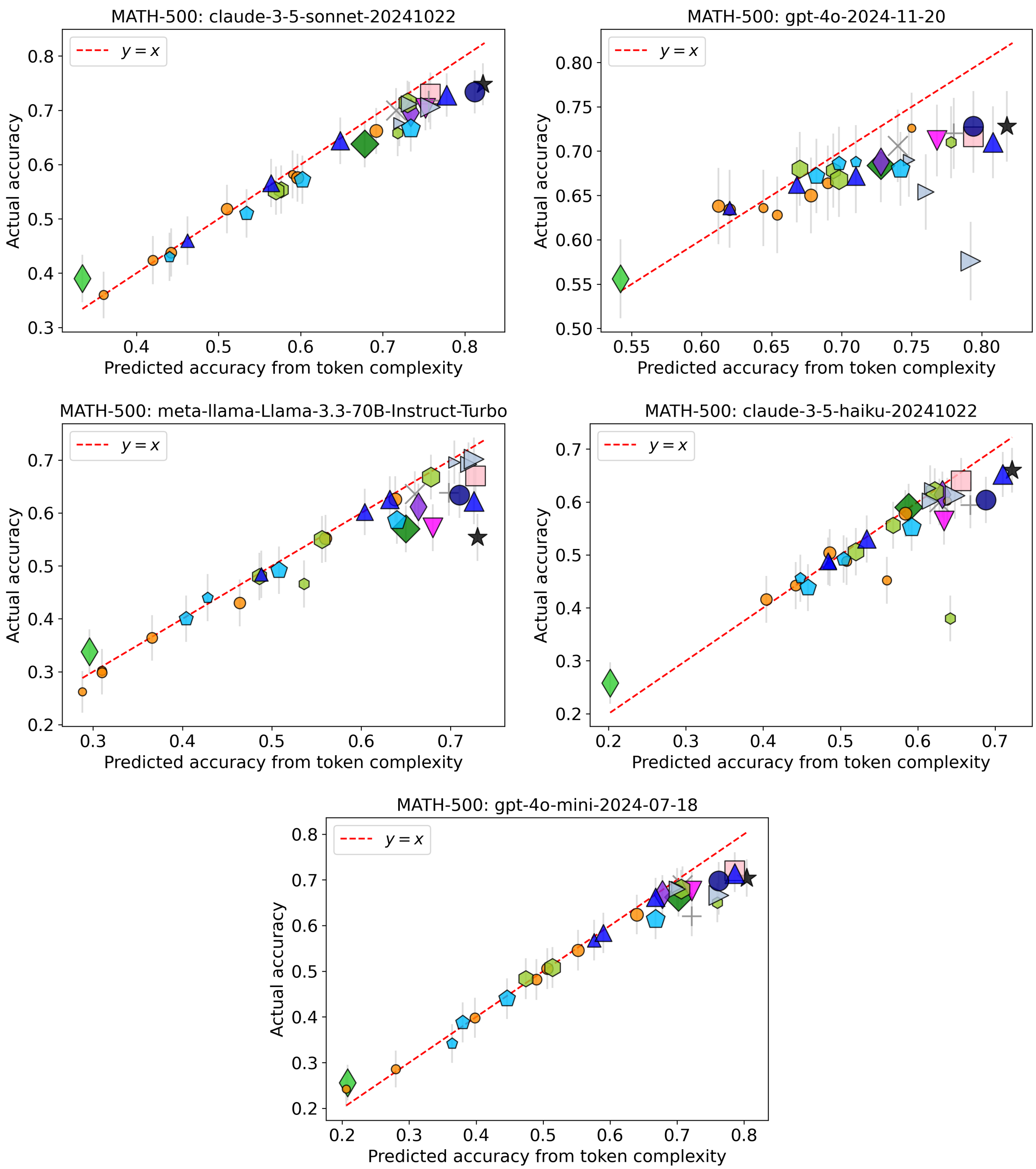

Figure 8: Actual vs Predicted Accuracy for MATH-500

H. Routing Performance on other benchmarks

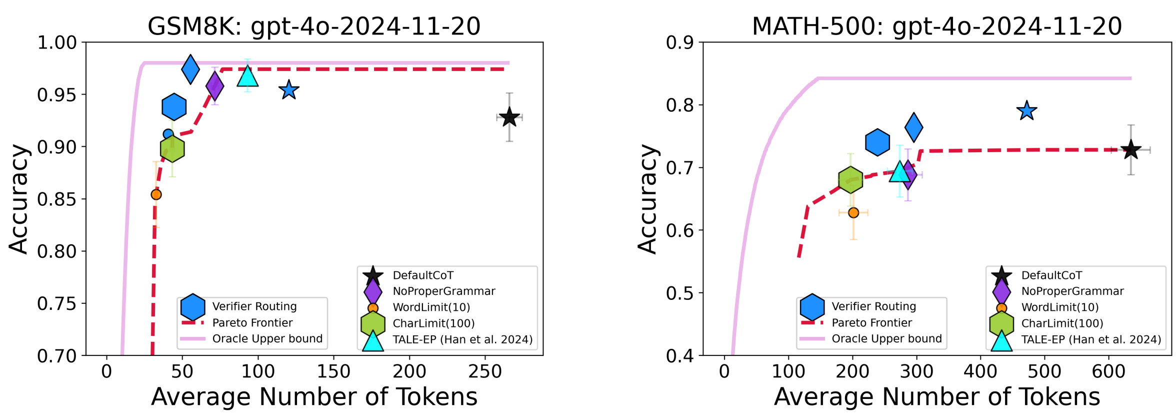

Figure 9: Performance of prompt routing on MATH-500 and GSM8K