Token-Budget-Aware LLM Reasoning

Tingxu Han[*1], **Zhenting Wang**[*†2], **Chunrong Fang**[‡1],
Shiyu Zhao[2], **Shiqing Ma**[3], **Zhenyu Chen**[1]
[1]Nanjing University [2]Rutgers University [3]UMass Amherst

Abstract

Reasoning is critical for large language models (LLMs) to excel in a wide range of tasks. While methods like Chain-of-Thought (CoT) reasoning and enhance LLM performance by decomposing problems into intermediate steps, they also incur significant overhead in token usage, leading to increased costs. We find that the reasoning process of current LLMs is unnecessarily lengthy and it can be compressed by including a reasonable token budget in the prompt, but the choice of token budget plays a crucial role in the actual compression effectiveness. We then propose a token-budget-aware LLM reasoning framework that dynamically adjusts the number of reasoning tokens based on the reasoning complexity of each problem. Experiments show that our method effectively reduces token costs in CoT reasoning with only a slight performance reduction, offering a practical solution to balance efficiency and accuracy in LLM reasoning. Code: https://github.com/GeniusHTX/TALE[1].

"It is not enough to have a good mind; the main thing is to use it well."

– René Descartes

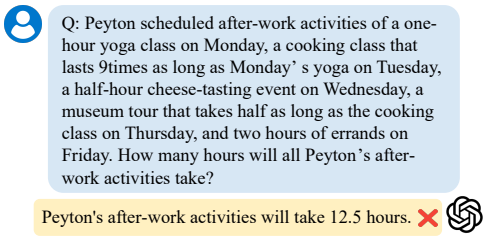
1 Introduction

Reasoning plays a crucial role in enabling large language models (LLM) to perform effectively across a wide range of tasks (Zhou et al., 2022; Hao et al., 2023, 2024a; Jin et al., 2024a; Wang et al., 2024b, 2025). A variety of methods have been proposed to enhance the reasoning capabilities of large language models (Suzgun et al., 2022; Wang et al., 2023; Feng et al., 2023; Xie et al., 2024). Among these, Chain-of-Thought (CoT) (Wei et al., 2022) is the most representative and widely adopted approach. It enhances the reliability of the model's answers by guiding large language models with the prompt "Let's think step by step", encouraging them to decompose the problem into intermediate steps and solve each before arriving at the final answer. Figure 1a and Figure 1b illustrate an intuitive example. Observe that without CoT, the LLM produces incorrect answers to the question. With a CoT-enhanced prompt, the LLM systematically breaks the question into multiple steps and reasons through each step sequentially. By addressing each step incrementally, the LLM eventually arrives at the correct answer. Recent reasoning models, such as OpenAI O1 (OpenAI, 2024c) and DeepSeek R1 (Guo et al., 2025), integrate CoT into their design. Notably, these models can perform CoT reasoning even without explicit prompting.

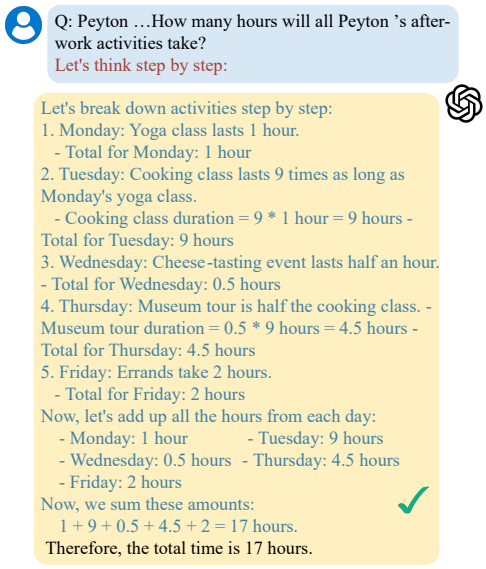
Although reasoning enhancement approaches such as CoT impressively improve LLM performance, they produce substantial additional overhead, specifically in the form of the increased number of tokens produced (Wei et al., 2022; Feng et al., 2023; Yao et al., 2024a; Jin et al., 2024b). As shown in Figure 1b, the answer to prompt with CoT has notably higher token costs due to the detailed intermediate reasoning steps included in the output. Such high token costs can lead to significant expenses, including increased computational resource usage and longer running times during the LLM inference, ultimately resulting in significant additional monetary and energy costs.

This raises an important question: *"Is the reasoning process of current LLMs unnecessarily lengthy, and how can it be compressed?"* Nayab et al. (2024) demonstrate that LLM has the potential to follow a length constraint in the prompt. Building on this, we find that *including a* ***token budget*** *(see Table 1) in the prompts is a promising approach to compressing the CoT reasoning tokens. However, the choice of token budget plays a crucial role in*

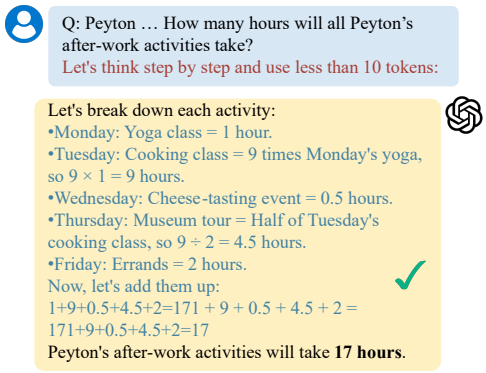
[*]Equal Contribution.
[†]Start the project and propose the idea.
[‡]Corresponding Author.
[1]Also available at https://www.gitlink.org.cn/txhan/TALE



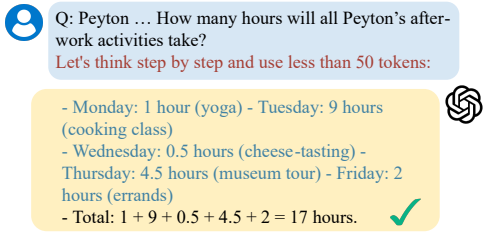
(a) Direct answering (15 output tokens).



(b) Vanilla CoT (258 output tokens).



(c) CoT with an unreasonable budget (157 output tokens).



(d) CoT with an reasonable budget (86 output tokens).

Figure 1: Examples of different problem solving paradigms. The reasoning processes are highlighted. We conduct this experiment on GPT-4o-mini.

the actual compression effectiveness. For example, Figure 1d illustrates that including a reasonable token budget (e.g., 50 tokens in this case) in the instructions reduces the token cost in the chain-of-thought (CoT) process from 258 output tokens to 86 output tokens, while still enabling the LLM to arrive at the correct answer. However, when the token budget is set to a different smaller value (e.g., 10 tokens), the output token reduction is less effective, resulting in 157 output tokens—nearly twice as many as with a 50-token budget. In other words, when the token budget is relatively small, LLMs often fail to follow the given token budget. In such cases, the actual token usage significantly exceeds the given budget—even much larger than the token costs with larger token budgets. We refer to this phenomenon as the "Token Elasticity" in the CoT process with token budgeting. To address this, the optimal token budget for a specific LLM and a particular question can be searched by gradually reducing the budget specified in the prompt, identifying the smallest token budget that achieves both the correct answer and the lowest actual token cost.

Based on the above observations and analysis, we propose a token-budget-aware LLM reasoning framework that dynamically adjusts the number of reasoning tokens based on the reasoning complexity of each problem. We call our method TALE (Token-Budget-Aware LLM rEasoning), which includes two implementations: token budget estimation and prompting (TALE-EP) and token budget awareness internalization via post-training (TALE-PT). TALE-EP estimates a reasonable token budget for each problem using zero-shot prompting and incorporates it into the reasoning process, while TALE-PT internalizes token-budget awareness through post-training, enabling the LLM to generate more token-efficient responses without explicit token constraints in the prompt. We discuss both implementations in Section 5. Experiment results show that TALE significantly reduces token costs in LLM chain-of-thought (CoT) reasoning while largely maintaining answer correctness. On average, TALE-EP achieves a 67% reduction in token usage while maintaining accuracy with less than a 3% decrease. TALE-PT cuts token usage by around 50% compared to Vanilla CoT and achieves competitive performance.

2 Related Work

LLM Reasoning. Reasoning in LLMs has seen substantial advancements through techniques that generate intermediate steps, enabling more accurate and effective performance across diverse domains (Wu et al., 2022; Yang et al., 2022; Zhou et al., 2022; Sun et al., 2024; OpenAI, 2024c). Var-

ious LLM reasoning techniques are proposed to improve the LLM performance. Chen et al. (2024) formulates reasoning as sampling from a latent distribution and optimizing it via variational approaches. Ho et al. (2022) utilizes LLM as reasoning teachers, improving the reasoning abilities of smaller models through knowledge distillation. Among them, Chain-of-Thought (CoT) prompting has emerged as a key technique for improving LLM reasoning by breaking problems into intermediate steps, enabling better performance on multiple tasks (Wei et al., 2022; Lyu et al., 2023; Li et al., 2023; Feng et al., 2024). Extensions of CoT include self-consistency, which aggregates multiple reasoning paths to improve robustness (Wang et al., 2022), and Tree-of-Thoughts, which explores reasoning steps in a tree-like structure for more complex tasks (Yao et al., 2024b). Reflexion introduces iterative refinement, where the model critiques and updates its intermediate steps (Shinn et al., 2024).

Token Cost of LLM. Although the above methods enhance reasoning accuracy, they often increase token usages, posing challenges to efficiency (Wang et al., 2024a; Chiang and Lee, 2024; Bhargava et al., 2023). Consequently, it is important to mitigate token consumption while maintaining the model performance. To address this issue, Li et al. (2021) introduces a multi-hop processing technique designed to filter out irrelevant reasoning. While effective, this approach is limited to traditional neural networks, such as PALM (Bi et al., 2020), and lacks adaptability to large language models (LLMs). Speculative decoding (Leviathan et al., 2023) aims to accelerate decoding by generating drafts using smaller models and verifying them with larger models, which is over-dependent on the alternative small approximation model. LLM routing (Ding et al., 2024) queries to different LLMs based on quality-cost trade-offs, but it cannot reduce the token usage on the specific LLM for a given query. Zheng et al. (2024) aims to improve LLM inference speed by predicting response lengths and applying a scheduling algorithm to enhance efficiency. However, it is constrained to scheduling level, and it does not reduce the actual token costs. Hao et al. (2024b) reduces token usage by substituting decoded text tokens with continuous latent tokens. However, its application is currently restricted to small-scale, early language models like GPT-2 (Radford et al., 2019). Additionally, it significantly impacts reasoning accuracy, resulting in over a 20% relative accuracy reduction on benchmarks such as GSM8K (Cobbe et al., 2021).

Table 1: Illustrations of the vanilla CoT prompt and the token-budget-aware prompt.

Prompt method	Content
Vanilla CoT	Let's think step by step:
CoT with Token Budget	Let's think step by step and use less than budget tokens:
Example	Let's think step by step and use less than 50 tokens:

3 Token Redundancy in LLM Reasoning

Token Budget. Previous research (Nayab et al., 2024) demonstrates that LLM has the potential to follow a length constraint in the prompt. Table 1 shows the difference between the vanilla CoT and the CoT with token budget. For instance, by including a token budget (50 tokens) within the prompt, as illustrated in Figure 1d, the LLM adjusts the length of its output (86 output tokens), trying to align with the specified budget. This indicates that LLMs have a certain capability in following prompts with an explicit token budget.

Token Redundancy Phenomenon. We find that providing a reasonable token budget can significantly reduce the token cost during reasoning. As shown in Figure 1d, including a token budget in the instructions reduces the token cost in the chain-of-thought (CoT) process by several times, but the LLM still gets the correct answer. Our results in Figure 2 and Table 3 also confirm there are a large number of redundant tokens in the reasoning process of the state-of-the-art LLMs.

Causes of Token Redundancy in LLM Reasoning. A possible explanation for this token redundancy is that during the post-training phase, such as the RLHF process (Ouyang et al., 2022), annotators might favor more detailed responses from LLMs, marking them as preferred. As a result, the model learns to associate longer, more detailed responses with alignment to human preferences and tends to produce such outputs during reasoning. However, in many scenarios, we primarily need LLMs to provide the correct answer and make accurate decisions, rather than elaborate extensively with detailed explanations. This motivates the need to eliminate redundant tokens in the LLM reasoning process in many cases.

4 Searching Optimal Token Budget

As demonstrated in Figure 1, different token budgets have different effects. Therefore, it is natural to investigate the following question: "*How to search*