

When More is Less: Understanding Chain-of-Thought Length in LLMs

Yuyang Wu*
Peking University

Yifei Wang*
MIT

Ziyu Ye
University of Chicago

Tianqi Du
Peking University

Stefanie Jegelka
TUM[†] and MIT[‡]

Yisen Wang[§]
Peking University

Abstract

Large Language Models (LLMs) employ Chain-of-Thought (CoT) reasoning to deconstruct complex problems. While longer CoTs are often presumed superior, this paper challenges that notion, arguing that **longer is not always better**. Drawing on combined evidence from real-world observations, controlled experiments, and theoretical analysis, we demonstrate that task accuracy typically follows an inverted U-shaped curve with CoT length, where performance initially improves but eventually decreases as the number of CoT steps increases. With controlled experiments, we further uncover the **scaling behaviors of the optimal CoT length**: it increases with task difficulty but decreases with model capability, exposing an inherent **simplicity bias** where more capable models favor shorter, more efficient CoT reasoning. This bias is also evident in Reinforcement Learning (RL) training, where models gravitate towards shorter CoTs as their accuracy improves. To have a deep understanding of these dynamics, we establish a simple theoretical model that formally proves these phenomena, including the optimal length’s scaling laws and the emergence of simplicity bias during RL. Guided by this framework, we demonstrate significant practical benefits from training with optimally-lengthed CoTs and employing length-aware filtering at inference. These findings offer both a principled understanding of the "overthinking" phenomenon and multiple practical guidelines for CoT calibration, enabling LLMs to achieve optimal reasoning performance with adaptive CoTs tailored to task complexity and model capability.

1 Introduction

"Everything should be made as simple as possible, but not simpler." — Albert Einstein

Large language models (LLMs) have demonstrated impressive capabilities in solving complex reasoning tasks [3, 36]. A key technique for its success is Chain-of-Thought (CoT) reasoning [38]. By generating explicit intermediate reasoning steps, CoT allows models to break down complex problems into simpler, more manageable sub-problems, akin to a divide-and-conquer strategy [44].

A common intuition, supported by some research [12, 20], is that longer and more detailed CoT processes generally lead to better performance, especially for difficult tasks. Meanwhile, recent observations also suggest that concise CoTs can sometimes be effective, albeit with potential performance

*Equal Contribution

[†]School of CIT, MCML, MDSI

[‡]EECS and CSAIL

[§]Corresponding Author: Yisen Wang (yisen.wang@pku.edu.cn)

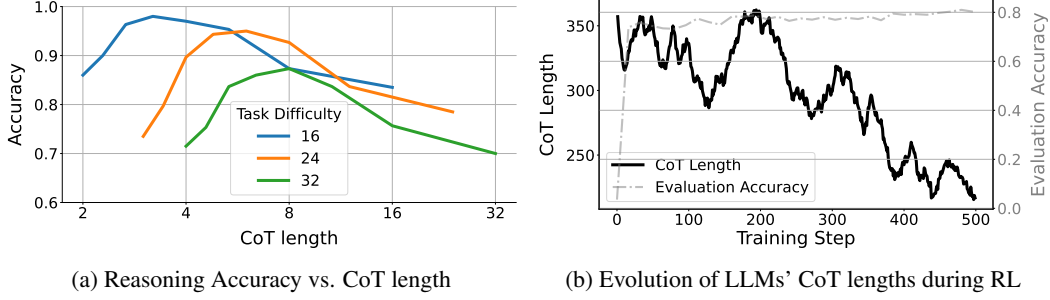


Figure 1: (a) The performance of a 6-layer GPT2 model (Section 3) follows inverted U-shaped curves on arithmetic tasks at different difficulty levels. As task difficulty increases, the accuracy peak progressively shifts toward longer CoT lengths. (b) As RL training progresses and model accuracy on reasoning tasks improves, the average length of the generated Chain-of-Thought can decrease. This hints at the model learning more efficient, concise reasoning paths (*i.e.*, simplicity bias). We conduct this experiment using Qwen2.5-7B-Instruct trained with GRPO on the LeetCode-2K dataset.

trade-offs on complex problems [26]. This raises a crucial question: does reasoning performance consistently improve as CoTs grow longer and longer?

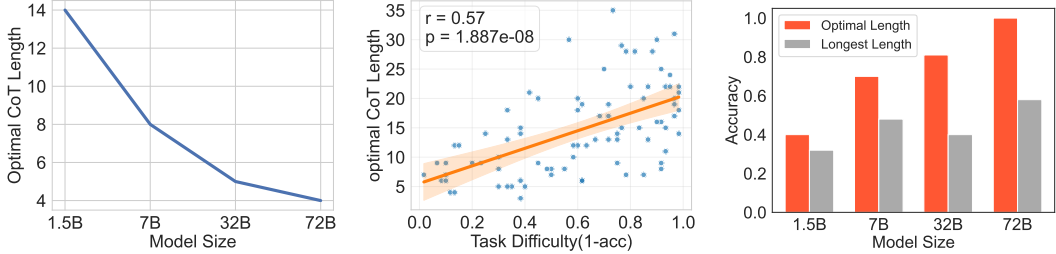
In this paper, through a comprehensive combination of evidence from theoretical analysis, controlled synthetic experiments, and real-world observations, we show that for CoT length, **longer is not always better**. As illustrated by the trend in Figure 1a, when plotting task accuracy against measures related to the CoT length, performance typically follows an **inverted U-shaped** curve. Performance initially improves as the CoT appropriately decomposes the task, but eventually deteriorates if the CoT becomes excessively long (increasing error accumulation) or too short (steps are too complex). This indicates the existence of an **optimal CoT length** that balances these competing factors.

Further, we discover scaling behaviors of this optimal CoT length with respect to model capability and task difficulty: harder tasks tend to have longer optimal CoTs, while more capable models often achieve peak performance with shorter optimal CoTs. This latter point interestingly implies an inherent **simplicity bias** in LLM reasoning, where models favor shorter, more efficient reasoning paths as their capabilities increase. Moreover, we observe this simplicity bias during LLMs’ reinforcement learning (RL) training. As shown in Figure 1b, RL-trained models exhibit a gradual shift towards using shorter CoTs compared to the base model, indicating an acquired preference for shorter CoTs as a result of the simplicity bias of optimal CoT length. This surprising phenomenon parallels humans’ natural preference for simplest possible reasoning processes, as evident in Einstein’s quote.

To gain a deeper understanding of the rise of optimal CoT length and its simplicity bias, we focus on a controlled study using a synthetic arithmetic task that allows us to ablate nuanced factors present in practical LLM training. In this controlled setting, we not only successfully replicate these phenomena but also theoretically derive the existence of the optimal CoT length and its scaling behaviors with respect to task complexity and model capability. Intuitively, task decomposition into more steps yields easier subtask but also accumulate errors exponentially, leading to an optimal tradeoff at an intermediate CoT length. Notably, this theory also explains the emergence of the simplicity bias as observed during RL training. Thus, although simple, our theory provides valuable characterization of LLMs’ behaviors during CoT. Translating this understanding into practice, we show significant benefits from training with optimally-lengthed CoTs and employing *Length-aware Vote* to filter out excessively long CoTs at inference.

To summarize, this paper makes the following main contributions:

- We demonstrate the existence of an optimal CoT length and the simplicity bias of CoT on both real-world LLMs (Section 2) and synthetic arithmetic experiments (Section 3).
- We establish a theoretical model of CoT that allows to formally characterize and prove the existence of an optimal CoT length as well as its scaling laws and simplicity bias (Section 4).
- We explore the implications of these findings, showing how training with optimal-length CoT data can significantly boost performance, and how filtering excessively long CoTs with entropy measures can benefit reasoning performance at inference (Section 5).



(a) Optimal CoT length vs. Model size (Qwen2.5 series). (b) Optimal CoT length vs. Task difficulty (with the 1.5B model). (c) Optimal vs. Longest CoT length accuracy on MATH Level 5.

Figure 2: Real-world CoT length observations. (a) Larger models tend to achieve optimal performance with shorter CoTs. (b) More difficult tasks (as measured by lower accuracy on the x-axis) tend to require longer optimal CoTs (with a positive correlation of significance $p \ll 0.05$). (c) Accuracy for CoTs of optimal length is significantly higher than that of the longest CoTs.

Our findings offer a fresh perspective for calibrating CoT generation, moving beyond the assumption that longer is always better. By understanding and adapting to the optimal CoT length, we can develop LLMs that reason more effectively, avoiding both underthinking and counterproductive overthinking.

2 Optimal CoT Length and Simplicity Bias in Real-World LLMs

To ground our investigation in practical scenarios, we first explore the relationship between CoT length and reasoning performance using publicly available LLMs.

2.1 Scaling Behaviors of Optimal CoT Length in Real-World LLMs

Setup. To assess how model capability interacts with CoT length. We evaluate Qwen2.5 series of Instruct models [27] on Level 5 questions in MATH dataset composed of challenging competition mathematics problems [18]. For each question, we generate 60 solutions with as much variation in length as possible. The CoT length is determined by the number of intermediate reasoning steps generated by the model. The optimal CoT length is the one that yields the highest average accuracy. See Appendix C for additional experiments (MMLU STEM dataset [17], different models) and implementation details on step segmentation and solution length control.

Optimal Length Decreases with Stronger Model Capabilities: For each model, we randomly select 30 questions since our focus lies in exploring different lengths of solutions for the same problem rather than evaluating the whole dataset. As depicted in Figure 2a, there is a clear trend where the optimal CoT length decreases as the model size increases. For instance, the optimal length shifts from 14 steps for the 1.5B parameter model to 4 steps for the 72B parameter model. This suggests that more capable models can consolidate reasoning into fewer, more potent steps, aligning with the Simplicity Bias concept where stronger models prefer shorter effective paths.

Optimal Length Grows with Harder Tasks: We also investigate how task difficulty influences the optimal CoT length. For this, we consider 100 randomly selected questions and compute the accuracy of an LLM on each question from 60 sampled solutions. We use $(1 - \text{accuracy})$ on these questions as a proxy for the difficulty. Figure 2b shows a statistically significant positive correlation (notably $p = 1 \times 10^{-8} \ll 0.05$) between task difficulty and the optimal CoT length of Qwen1.5B-Instruct model. This indicates that more challenging problems will significantly benefit from a longer CoT with more extended decomposition steps. Similar trends for other models are provided in Appendix C.2.

Excessively Long CoTs Lead to Significant Degradation: The above scaling behaviors of CoT suggest that one should adaptively select the optimal CoT length w.r.t. the given model and task. Here, we illustrate the significance of this choice by compare the performance of using the optimal and the longest CoT lengths. As shown in Figure 2c, there is a large gap between the two that grows larger as the models become more capable. For a 72B model, the gap can be as large as 40% accuracy, showing great potential gains of adapting CoT length to attain optimal reasoning performance.

2.2 Simplicity Bias in Reinforcement Learning

A common belief in the ongoing development of advanced reasoning models is that reinforcement learning (RL) leads to more lengthy output in reasoning models. Nevertheless, recent studies [13] also revealed that RL-trained model behaviors remain largely depend on the base model. It is yet unclear what is the exact influence of RL training on CoT length. To have a clear understanding of this process, we monitor the evolution of CoT length during GRPO training [31], using LeetCode-2K [40] with Qwen2.5-7B-Instruct [27]. We refer readers to Appendix D for additional details and ablations. As shown in Figure 1b, through optimizing outcome rewards from model rollouts, the average response length of RL models can decrease as training converges. As a result, RL-trained model has shorter CoTs (on average) than the base model, indicating that RL has a *simplicity bias* that favors shorter answers instead of long answers.

3 A Controlled Study of CoT Length in Arithmetic Tasks

The observations from real-world LLMs in Section 2 suggest a complex interplay between CoT length, model capability, and task difficulty. However, real-world CoTs involve numerous uncontrolled variables (e.g., diverse reasoning strategies, planning, backtracking) and varying types of base model pre-training, making a precise mechanistic understanding challenging. To overcome these limitations and rigorously examine our hypotheses about optimal CoT length and Simplicity Bias, we develop a controlled experimental setup using synthetic arithmetic tasks.

3.1 Experimental Setup

Dataset: Our synthetic dataset consists of arithmetic problems involving only a sequence of addition operations. The inherent difficulty of a problem is quantified by the total number of addition operators, T . For any given problem with T operators, we generate multiple valid CoT solutions, differing in their length and granularity. The CoT length N , is the number of intermediate reasoning steps. Each step i in a CoT processes a certain number of operators, t_i . For simplicity in our controlled study, we structure solutions such that t_i is (approximately) constant for all steps in a given CoT, denoted as the step size t (operators per step), where $N \approx T/t$.

For example, consider an arithmetic problem like "1 + 2 + 3 + 4 + 5 + 6 + 7". This problem involves $T = 6$ addition operators. We can construct different CoT solutions for this problem:

- A *long CoT solution* might be designed to process $t = 1$ operator per step. This would result in $N = 6$ reasoning steps.

Problem: 1+2+3+4+5+6+7
Step 1: 1+2 = 3. (Remaining: 3+3+4+5+6+7)
Step 2: 3+3 = 6. (Remaining: 6+4+5+6+7)
...
Step 6: 21+7 = 28. (Final Answer)

- A *shorter CoT solution* for the same problem might process $t = 3$ operators per step. This would result in $N = 2$ reasoning steps.

Problem: 1+2+3+4+5+6+7
Step 1: 1+2+3+4 = 10. (Remaining: 10+5+6+7)
Step 2: 10+5+6+7 = 28. (Final Answer)

This dataset design is crucial as it allows us to systematically vary the CoT length (N) or the number of operators processed per step (t) for problems of a fixed total difficulty (T). This enables a focused study on how the structure of the reasoning process itself impacts performance. More discussion of problem definition, data format, CoT generation, and considerations for choosing task data formatting and task design, is provided in Appendix B.

Model and Training: We train GPT-2 models [28] of varying depths (number of layers), keeping other hyperparameters fixed. Model depth is known to be a significant factor representing model capabilities for reasoning tasks [43, 4]. Controlling this hyperparameter alone allows us to study the impact of model capability on optimal CoT length. Models are trained with CoT solutions that can be automatically synthesized for this task, with varying total operators T and CoT lengths N (or equivalently the step sizes t). For testing, we can guide the model to produce a CoT of a specific

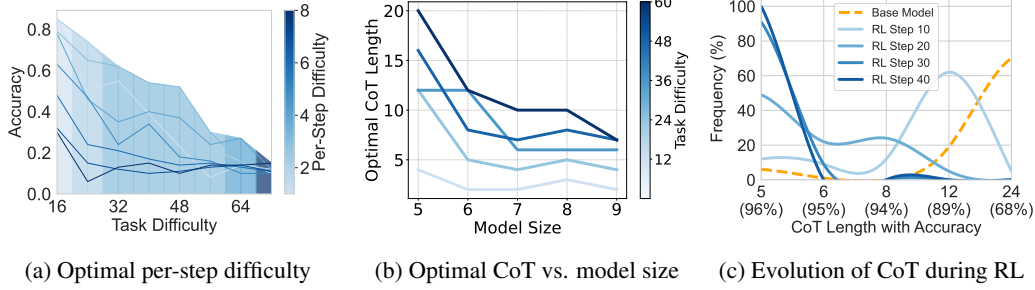


Figure 3: **CoT Behaviors in Synthetic Experiments:** (a) Each curve corresponds to a specific CoT strategy with fixed per-step difficulty. The color of the bar beneath each curve represents the optimal per-step difficulty (t) at each task difficulty. The progressively darker gradient colors indicates that harder tasks consistently favor higher per-step difficulty. (b) Change of the optimal CoT length with increasing model size across different task difficulty levels. As model size increases, the optimal CoT length decreases. For a fixed model size, harder tasks also exhibit longer optimal CoTs. (c) During RL training, the model policy gradually favors a shorter CoT that corresponds to the optimal length. length (e.g., by prompting with a control token indicating the desired number of operators t per step) or allow it to choose its preferred length. Further details are in Appendix E.

3.2 Scaling Laws of the Optimal CoT Length and and Practical Insights

Our controlled experiments not only corroborate the CoT behaviors observed in real-world scenarios but also allow for a more fine-grained analysis. These findings uncover several key scaling behaviors of the optimal CoT length that shed light into the practical designs of LLM reasoning.

I. Harder-Tasks’ CoTs Peak at Longer Lengths (Adaptive CoT Length Matters): Our synthetic experiments further confirm the existence of an optimal CoT length, which manifests itself as an inverted U-shaped performance curve when plotting accuracy against the number of reasoning steps, as shown in Figure 1a. This clearly indicates that both "underthinking" (CoT too short) and "overthinking" (CoT too long) are detrimental, underscoring the critical benefit of generating CoTs with adaptive lengths tailored to the problem’s demands. Moreover, we observe that the optimal CoT length shifts right as the task difficulty T gets larger, indicating that solving a harder task optimally requires a longer CoT (also observable numerically from Figure 3b). This suggests that a good reasoning model should be able to vary CoT lengths w.r.t. the overall task complexity.

II. Harder Tasks Peak at Harder Sub-tasks (Adaptive Per-Step Computation Helps): Figure 3a illustrates how the number of operators per step (t) impacts model accuracy across varying task difficulties (T). The envelope curve, tracing peak performance, reveals that as tasks become more challenging (larger T), optimal performance is often achieved by CoTs that involve more complex computations *per step* (i.e., a larger optimal t^*). This suggests that for harder problems, simply increasing the number of simple steps may not be as effective as increasing the complexity of each sub-task the model tackles within the CoT. Current LLMs with fixed Transformer layers have limited intrinsic ability to adapt their per-step computational depth for different sub-tasks. This implies that their reasoning strategy might remain suboptimal. In contrast, recent advancements like looped Transformers, which enable adaptive recurrent depth [14, 8], could offer a more promising avenue for dynamically adjusting per-step computation to align with this observed need, potentially leading to better reasoning performance.

III. Stronger Models Achieve Optimal Performance with Shorter CoTs (Model-Aware CoT Data Matter): We also examine how model capability (number of layers) influences the optimal CoT length. Figure 3b indicates that, across different task complexities, the optimal number of CoT steps (N^*) consistently decreases as the model’s capability (number of layers) increases. This is because stronger models can effectively handle more complex sub-tasks in each step, thus requiring fewer overall steps to reach the solution optimally. This finding has significant implications for training data curation. It suggests that to achieve peak performance, models of different sizes or capabilities require CoT data tailored to their respective optimal per-step complexities. Current practices, such as using the same CoT datasets to train LLMs of varying sizes or directly distilling CoTs from large models to small ones without adapting complexity, may be suboptimal. For instance, a small model might struggle to learn effectively from overly complex CoT demonstrations designed for a larger

model. Our analysis advocates for training each model with CoT data of adaptive complexity, aligned with its specific capabilities, to help it reach its optimal reasoning performance.

IV. RL Training Converges to Optimal CoT Length (RL Calibrates Reasoning Behaviors):

As discussed in Section 2.2, RL training of LLMs leads to shorter CoT lengths. Our synthetic experiments further replicate this phenomenon. We take a GPT-2 model pre-trained on CoT solutions of equally mixed lengths for a task of difficulty $T = 24$ and apply RL using rule-based outcome rewards with PPO on VERL [30, 32]. Figure 3c shows the change of the sampled CoT lengths along RL: as training progresses, the model increasingly favors the CoT structure corresponding to the optimal length $N^* = 5$ that yields the peak accuracy (96%) on this task. This demonstrates that RL, by optimizing for task success, can implicitly guide the model’s CoT generation policy towards the optimal length regime, thereby exhibiting the *simplicity bias*. This offers a fresh perspective for understanding the benefits of RL in LLM training: even if the initial CoT data used for pre-training or supervised fine-tuning is suboptimal (e.g., misaligned with the model size or the task complexity), RL can help calibrate the model’s behavior towards generating more optimally-lengthy CoTs.

4 Theoretical Analysis: Why an Optimal CoT Length Exists

The empirical findings from both real-world and synthetic datasets consistently point to the existence of an optimal Chain-of-Thought (CoT) length. In this section, we provide a theoretical framework to explain this phenomenon, formalizing how factors like task decomposition and error accumulation interact to determine this optimal length, and how it scales with model capability and task difficulty. All proofs are deferred to Appendix G.

4.1 Theoretical Formulation

Akin to the arithmetic tasks we studied in Section 3, we use the following simple theory model to describe the CoT process.⁵ Let $N \in \mathbb{N}^+$ be the total number of steps in the CoT process. Let T denote the total number of operators in the given arithmetic task (a proxy for task difficulty). We assume that each CoT step consists of a sub-question q_i (e.g., $2 + 1 =$) and its answer as a_i (e.g., 3).

Definition 4.1 (CoT Process Probability). Given a task q with T total operators and a model θ , the probability of an N -step CoT that leads to a final answer a_{final} is:

$$P(a_{\text{final}}|q, \theta, N) = \prod_{i=1}^N \underbrace{P(q_i|H_{i-1}, q, \theta, N)}_{\text{sub-question}} \underbrace{P(a_i|q_i, H_{i-1}, q, \theta, N)}_{\text{sub-answer}},$$

where $H_k := [t_1, a_1, \dots, t_k, a_k]$ denotes the CoT history of the first k steps.

Let a_i^* denote the correct answer to subtask q_i and q_i^* denote the unique correct sub-question for simplicity. To estimate the final accuracy $A(N) = P(a_N = a_N^*|q, \theta)$, we need to estimate the sub-question accuracy $P(q_i = q_i^*|H_{i-1}, q, \theta)$ and the sub-answer accuracy $P(a_i = a_i^*|q_i, H_{i-1}, q, \theta)$.

For the **sub-question accuracy**, following experimental observation (in Appendix E.2), we assume that the error rate of generating each question q_i , denoted by $\sigma(T) \in [0, 1]$, is positively correlated with the total number of operators T . Intuitively, as the number of operators increases, extracting the correct subtask becomes more challenging. For the **sub-answer accuracy**, it is clear that when given subtask q_i , $P(a_i = a_i^*|q_i, H_{i-1}, q, \theta)$ is independent of the history reasoning steps H_{i-1} and is only influenced by the model θ and the difficulty of the subtask q_i . For each model, we define its capability M based on the reasoning boundary [5], e.g., the maximum number of operators the model can directly solve per step; thus, a stronger model has a larger M . We define the error rate of each subtask answer as $E(N, M, T) \in [0, 1]$.

Proposition 4.2. *The total accuracy of N -step reasoning is*

$$A(N) = P(a_{\text{final}} = a_{\text{final}}^*|q, \theta, N) = \alpha ((1 - E(N, M, T))(1 - \sigma(T)))^N, \quad (1)$$

where α denotes a constant value independent of N .

⁵Note that we do not explicitly model various cognitive reasoning behaviors (reflection, verification, backtracking) but instead regard them as one of the many ways that one can decompose a task into subtasks to ease problem solving, which can be understood as a part of our task decomposition formulation in a general sense.

Proposition 4.2 establishes the quantitative relationship between CoT’s final performance A and reasoning length N . Once we obtain estimates for $E(N, M, T)$ and $\sigma(T)$, we can determine the optimal CoT length N^* as a function of the model capability M and task complexity T .

Simple Case with Linear Error. To gain intuition, we first consider a simple case where the sub-question error scales linearly with T , i.e., $\sigma(T) = T/C$, where C is a constant representing the maximum task difficulty models are trained to handle. Throughout this analysis, we assume $\sigma(T) \leq 0.9$ to restrict our discussion to tasks that are within the model’s training regime. Otherwise, it would be unreasonable to claim the model has learned to solve such problems. We also assume the sub-answer error rate scales linearly with harder tasks, fewer steps, and weaker models as $E(N, M, T) = (T/N)/M = T/(NM)$. Under these simplified conditions, we can derive the following closed-form expression for the optimal CoT length.

Theorem 4.3 (Optimal CoT Length). *For a given model capability M and task difficulty T , the total accuracy $A(N) = \alpha[(1 - T/C) \cdot (1 - T/(NM))]^N$ (Eq. (1)) initially increases and then decreases as N increases (forming an inverted U-shape). Thus, there exists an optimal CoT length:*

$$N^*(M, T) = \frac{TZ}{M(Z + 1)}, \quad (2)$$

that maximizes $A(N)$, where $Z = W_{-1}(-1 - T/(Ce))$, and $W_{-1}(x)$ is the smaller real branch of the Lambert W function satisfying $we^w = x$, and e is the natural number.

This theorem formally establishes the inverted U-shaped curve and provides an explicit form for N^* . From this, we can formally prove the first three scaling behaviors characterized in Section 3.2.

Corollary 4.4 (Scaling laws of Optimal CoT Length). *Based on Theorem 4.3, one can derive:*

- $N^*(M, T)$ increases monotonically with T , i.e., harder tasks require more reasoning steps to attain the optimal performance.
- The optimal number of operators per step $t^* = T/N^*(M, T) = M(1 + 1/Z)$ increases monotonically with T . This aligns with the envelope curve result (Figure 3a).
- $N^*(M, T)$ decreases monotonically with M , i.e., more capable models require fewer reasoning steps to attain the optimal performance, reflecting the simplicity bias.

Extension to Broader Scenarios. Here, we adopt a simple linear model to facilitate intuitive understanding. However, this analysis can be extended to more general settings, including general error functions (only with mild assumptions of monotonicity and convexity) and stochastic error models, where each subtask may exhibit a different error rate. These extensions introduce additional technical subtleties but follow the same underlying principles. We defer this part to Appendix F.

4.2 Why does RL Exhibit Simplicity Bias?

The analysis above also provides a natural understanding of RL’s simplicity bias (Section 3.2). As in the arithmetic task, we generate samples within a finite discrete action space $\mathcal{A} = \{N_1, N_2, \dots, N_k\}$ during RL that receive binary outcome rewards. This reduces to a stateless bandit: each N_i yields reward $r \in \{0, 1\}$ with probability $A(N_i)$ (from Proposition 4.2). Let us parameterize a softmax policy $\pi_\theta(N_i) = \frac{e^{\theta_i}}{\sum_j e^{\theta_j}}$, and define the RL objective as $J(\theta) = \sum_{i=1}^k \pi_\theta(N_i) A(N_i)$. As a result, the policy-gradient becomes $\nabla_{\theta_i} J = \sum_{j=1}^k A(N_j) \pi_\theta(N_j) (\delta_{ij} - \pi_\theta(N_i))$.

Corollary 4.5 (RL Converges to Optimal CoT Length). *For gradient ascent on $J(\theta)$ with sufficiently small step size, the policy converges to a deterministic policy $\pi_\theta(N_i) = 1$ iff $i = \arg \max_j A(N_j)$. Thus, RL training converges to the optimal CoT length $N^* = \arg \max_{N \in \mathcal{A}} A(N)$.*

This corollary shows that RL will automatically discover the optimal length (usually shorter length) through optimizing the reward function and exhibit a decreasing CoT length as in the simplicity bias phenomenon. In this way, our theory offers an explanation of the optimal CoT length, its scaling behavior and RL’s simplicity bias within a unified framework.

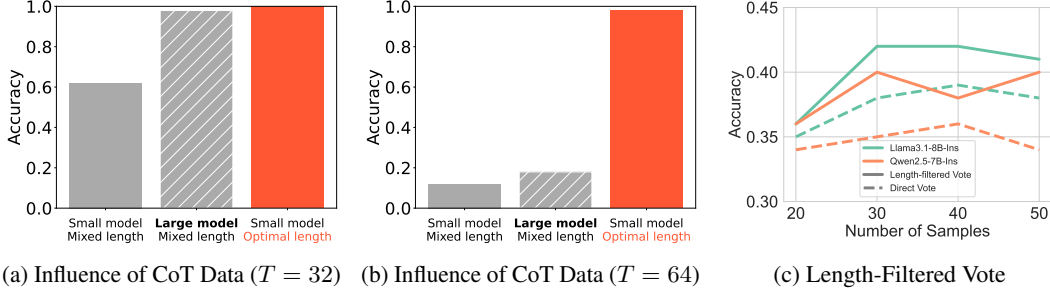


Figure 4: (a) and (b) compare model performance under different pretraining data distributions: Mixed Length (uniform over all lengths) vs. Optimal Length (only optimal-length solutions). Despite its smaller size, the small (6 layer) model trained on optimal-length data outperforms the large (9 layer) model trained on mixed-length data, with the performance gap widening as task difficulty increases. (c) Our Length-Filtered Vote method consistently outperforms vanilla majority vote on the GPQA dataset, maintaining strong performance even as the number of samples increases.

5 Practical Applications of Optimal CoT Length

Guided by the understanding above, in this section, we illustrate via some proof-of-concept experiments that adapting LLM training and inference configurations to the optimal CoT length can improve the model’s reasoning performance.

5.1 Training with Data of Optimal CoT Length

Training with Optimal-Length CoT Data: The existence of an adaptive, optimal CoT length suggests that one should design the CoT training data adaptively to fully optimize the model’s reasoning performance. To examine the influence of the CoT length of the training data, we train a model on a specialized dataset that contains CoT solutions with lengths known to be optimal for the given model size and task difficulty (T). We compare this model against a baseline model trained on a dataset of CoT solutions with uniformly distributed step lengths t . During testing, models were allowed to freely choose their CoT strategy.

Results. As shown in Figures 4a and 4b, the model trained on optimal-length CoTs significantly outperforms the models trained on mixed-length solutions. Remarkably, a smaller model (e.g., 6 layers) trained on optimal-length data can even outperform a larger model (e.g., 9 layers) trained on randomly chosen CoT lengths. This proof-of-concept experiment underscores the critical influence of the suitability of the CoT length in training data for the model. While it is generally hard to exactly estimate optimal CoT lengths in real-world problems, our theoretical and empirical studies provide valuable guidelines for a coarse estimate. We leave more in-depth studies to future work.

5.2 Adaptive Length-Filtered Vote at Inference Time

The observation that CoTs of optimal length yield higher accuracy suggests that inference-time strategies could benefit from this insight. Standard approaches like majority voting over multiple sampled CoTs, such as self-consistency [37], treat all valid reasoning paths equally, regardless of their length. However, paths that are too short (underthinking) or too long (overthinking and error-prone) may contribute noisy or incorrect answers to the voting pool.

Inspired by our findings, we propose **Length-Filtered Vote**, an adaptive method that enhances standard majority voting by preferentially weighting or exclusively considering answers derived from CoTs whose lengths fall within a proper range. Specifically, in majority vote, given a model f_θ , a question q , a ground truth answer a^* , we first sample a set of answer candidates $c_1, \dots, c_n \stackrel{i.i.d.}{\sim} f_\theta(q)$ independently. After that, instead of a direct vote, we group the answers by their corresponding CoT length $\ell(c_i)$ (discussed in Appendix C) into groups with equal bin size D (by default, we set $D = 2$), denoted as $\{L_j\}_{j=1}^m$. As our theory suggests that the prediction accuracy is peaked around a certain range of CoT length, we identify such groups through the prediction uncertainty of the answers within each group, based on the intuition that lower uncertainty implies better predictions. Specifically, we

calculate the Shannon entropy $H(L_i)$ of the final answers given by the CoT chains in each group L_i . We use a majority vote only for the K (by default, we set $K = 3$) groups with the smallest entropy. A detailed description of the algorithm is in Appendix H.

Results. We evaluate the proposed method against vanilla majority vote (i.e., self-consistency [37]) on a randomly chosen subset of 100 questions from the GPQA dataset [29], a more challenging collection of multiple-choice questions. The results in Figure 4c show that our filtered vote consistently outperforms vanilla majority vote at different sample numbers and shows little performance degradation as the sample number increases. This further underlines the importance of considering CoT length in the reasoning process.

6 Related Work

Chain-of-Thought for LLM Reasoning. CoT has become a core technique for LLMs to solve complex reasoning tasks by generating intermediate steps [38]. Numerous variants arise to enhance CoT reasoning with more structural substeps, such as least-to-most prompting [45], tree of Thoughts [42], and divide-and-conquer methods [44, 25]. These methods fundamentally treat CoT as a framework for task decomposition and subtask solving that falls in our analysis in Section 4.

Overthinking in CoT Reasoning. With the rise of powerful reasoning models like OpenAI o1, scaling test-time compute with long CoT has gained prominence [33, 7, 39, 2]. These studies often suggest that more computation like longer CoT can lead to better results. However, this is not always true. With similar interests as ours, a few concurrent works also investigated the “overthinking” phenomenon [6] where reasoning models generate excessively long CoTs for simple problems and proposed some mitigation strategies [16, 23, 24, 34]. Our analysis not only reveals the inverted-U curve of CoT length and the existence of optimal CoT length, but also provides a in-depth understanding on the scaling behaviors and simplicity bias of the optimal CoT length, as supported by both controlled experiments and theoretical analysis. This establishes a systematic explanation of overthinking and points out principled guidelines for better CoT designs.

Simplicity Bias and Occam’s Razor in Machine Learning. The simplicity bias of CoT identified in our work resonates with broader principles like Occam’s Razor, which favors simpler explanations or models. In machine learning, a ‘simplicity bias’ often refers to neural networks learning simpler functions first or being biased towards solutions with lower intrinsic complexity [1, 19]. Our findings extend this understanding to the realm of generated reasoning paths: we reveal that even structured, multi-step reasoning processes like CoT, as produced by LLMs, exhibit such simplicity bias by favoring concise reasoning paths, particularly as model capability increases.

Theoretical Understanding of CoT. Numerous studies aim to theoretically formalize the Chain-of-Thought (CoT) process and understand its effectiveness. They include analyzing CoT’s computational advantages via circuit complexity [11, 22], demonstrating how coherent reasoning paths enhance error correction and accuracy [10], and quantifying step-wise information gain from an information-theoretic standpoint [35]. Further research has shown that detailed CoT improves learning stability by affecting gradient dynamics [21], while controlled synthetic experiments have helped uncover underlying problem-solving mechanisms in LLMs [43]. Distinct from these varied theoretical explorations, our theory characterizes how CoT length influences final performance and explains its scaling behaviors through the interplay of task decomposition and error accumulation. Furthermore, our findings on CoT scaling behaviors and the consequent need for model-specific CoT structures (as discussed in Section 3.2) resonate with the concept of algorithmic alignment [41], which suggests that models perform best when the problem structure aligns with their computation structure.

7 Conclusion

In this paper, we challenged the notion that longer Chain-of-Thought (CoT) processes are invariably superior, demonstrating through extensive experiments and theoretical analysis that CoT length and accuracy typically follow an inverted U-shaped curve, implying an optimal length that balances task decomposition against error accumulation. We discovered the simplicity bias of CoT, where more capable models prefer shorter effective reasoning paths, and formally derived scaling laws for this optimal length relative to model capability and task difficulty. Practically, we showed that reinforcement learning can guide models towards this optimal CoT length, that training on

optimally-lengthed CoTs boosts performance, and proposed "Length-Filtered Vote" as a promising inference strategy. Our work underscores the critical need to calibrate CoT length, moving beyond a one-size-fits-all approach towards a principled framework where LLMs adaptively choose the right amount of thought to optimize reasoning.

Acknowledgement

Yisen Wang was supported by National Key R&D Program of China (2022ZD0160300), National Natural Science Foundation of China (92370129, 62376010), and Beijing Nova Program (20230484344, 20240484642). Yifei Wang and Stefanie Jegelka were supported in part by the NSF AI Institute TILOS (NSF CCF-2112665), and an Alexander von Humboldt Professorship.

References

- [1] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR, 2017.
- [2] Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling, 2024. URL <https://arxiv.org/abs/2407.21787>.
- [3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf.
- [4] Lijie Chen, Binghui Peng, and Hongxun Wu. Theoretical limitations of multi-layer transformer, 2024. URL <https://arxiv.org/abs/2412.02975>.
- [5] Qiguang Chen, Libo Qin, Jiaqi WANG, Jingxuan Zhou, and Wanxiang Che. Unlocking the capabilities of thought: A reasoning boundary framework to quantify and optimize chain-of-thought. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=pC44UMwy2v>.
- [6] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do not think that much for $2+3=?$ on the overthinking of o1-like llms, 2024. URL <https://arxiv.org/abs/2412.21187>.
- [7] Yanxi Chen, Xuchen Pan, Yaliang Li, Bolin Ding, and Jingren Zhou. A simple and provable scaling law for the test-time compute of large language models, 2024. URL <https://arxiv.org/abs/2411.19477>.
- [8] Yilong Chen, Junyuan Shang, Zhenyu Zhang, Yanxi Xie, Jiawei Sheng, Tingwen Liu, Shuohuan Wang, Yu Sun, Hua Wu, and Haifeng Wang. Inner thinking transformer: Leveraging dynamic depth scaling to foster adaptive internal thinking, 2025. URL <https://arxiv.org/abs/2502.13842>.
- [9] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>.

- [10] Yingqian Cui, Pengfei He, Xianfeng Tang, Qi He, Chen Luo, Jiliang Tang, and Yue Xing. A theoretical understanding of chain-of-thought: Coherent reasoning and error-aware demonstration, 2024. URL <https://arxiv.org/abs/2410.16540>.
- [11] Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=qHrADgAdYu>.
- [12] Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning, 2023. URL <https://arxiv.org/abs/2210.00720>.
- [13] Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars, 2025. URL <https://arxiv.org/abs/2503.01307>.
- [14] Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*, 2025.
- [15] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [16] Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*, 2024.
- [17] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=d7KBjmI3GmQ>.
- [18] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL <https://arxiv.org/abs/2103.03874>.
- [19] Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *arXiv preprint arXiv:2103.10427*, 2021.
- [20] Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. The impact of reasoning step length on large language models. In *Annual Meeting of the Association for Computational Linguistics*, 2024. URL <https://api.semanticscholar.org/CorpusID:266902900>.
- [21] Ming Li, Yanhong Li, and Tianyi Zhou. What happened in llms layers when trained for fast vs. slow thinking: A gradient perspective, 2024. URL <https://arxiv.org/abs/2410.23743>.
- [22] Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems, 2024. URL <https://arxiv.org/abs/2402.12875>.
- [23] Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*, 2025.
- [24] Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. Cot-valve: Length-compressible chain-of-thought tuning. *arXiv preprint arXiv:2502.09601*, 2025.
- [25] Zijie Meng, Yan Zhang, Zhaopeng Feng, and Zuozhu Liu. Dcr: Divide-and-conquer reasoning for multi-choice question answering with llms, 2024. URL <https://arxiv.org/abs/2401.05190>.
- [26] Sania Nayab, Giulio Rossolini, Giorgio Buttazzo, Nicolamaria Manes, and Fabrizio Giacomelli. Concise thoughts: Impact of output length on llm reasoning and cost, 2024. URL <https://arxiv.org/abs/2407.19825>.

- [27] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL <https://arxiv.org/abs/2412.15115>.
- [28] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [29] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL <https://arxiv.org/abs/2311.12022>.
- [30] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL <https://arxiv.org/abs/1707.06347>.
- [31] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.
- [32] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys '25*, page 1279–1297. ACM, March 2025. doi: 10.1145/3689031.3696075. URL <http://dx.doi.org/10.1145/3689031.3696075>.
- [33] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL <https://arxiv.org/abs/2408.03314>.
- [34] Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.
- [35] Jean-Francois Ton, Muhammad Faaiz Taufiq, and Yang Liu. Understanding chain-of-thought in llms through information theory, 2024. URL <https://arxiv.org/abs/2411.11984>.
- [36] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothee Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- [37] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=1PL1NIMMrw>.
- [38] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022.
- [39] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Scaling inference computation: Compute-optimal inference for problem-solving with language models. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024. URL <https://openreview.net/forum?id=j7DZWSc8qu>.
- [40] Yunhui Xia, Wei Shen, Yan Wang, Jason Klein Liu, Huifeng Sun, Siyue Wu, Jian Hu, and Xiaolong Xu. Leetcodedataset: A temporal dataset for robust evaluation and efficient training of code llms, 2025. URL <https://arxiv.org/abs/2504.14655>.

- [41] Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. What can neural networks reason about? *arXiv preprint arXiv:1905.13211*, 2019.
- [42] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=5Xc1ecx01h>.
- [43] Tian Ye, Zicheng Xu, Yuanzhi Li, and Zeyuan Allen-Zhu. Physics of language models: Part 2.1, grade-school math and the hidden reasoning process, 2024. URL <https://arxiv.org/abs/2407.20311>.
- [44] Yizhou Zhang, Lun Du, Defu Cao, Qiang Fu, and Yan Liu. An examination on the effectiveness of divide-and-conquer prompting in large language models, 2024. URL <https://arxiv.org/abs/2402.05359>.
- [45] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=WZH7099tgfM>.

Appendix

Contents

| | | |
|----------|--|-----------|
| A | Limitations | 14 |
| B | Formal Definitions of Simplified Arithmetic Problem | 15 |
| B.1 | Problem Formulation | 15 |
| B.2 | Contrast to vanilla arithmetic problem | 16 |
| C | Supplementary Details on Real world Experiment for Optimal CoT Length | 16 |
| C.1 | Implementation Details | 16 |
| C.2 | More Experimental Results | 17 |
| D | Supplementary Details on Real World Experiment for RL Simplicity Bias | 19 |
| E | Additional Synthetic Experiment Details | 19 |
| E.1 | Training details | 19 |
| E.2 | Observation of subtask loss | 19 |
| F | Theoretical Results under Broader Scenarios | 19 |
| F.1 | General Errors | 19 |
| F.2 | Random Error | 20 |
| G | Proof | 20 |
| G.1 | Proof of Proposition 4.2 | 21 |
| G.2 | Proof of Theorem 4.3 | 21 |
| G.3 | Proof of Corollary 4.4 | 22 |
| G.4 | Proof of Theorem F.3 | 23 |
| G.5 | Proof of Theorem F.6 | 24 |
| G.6 | Proof of Corollary 4.5 | 24 |
| G.7 | Technical Lemmas | 25 |
| H | Pseudo-code of Length-filtered Vote | 27 |

A Limitations

This proof-of-concept study highlights the critical role of aligning Chain-of-Thought (CoT) length with model capability and task difficulty, particularly in training data. However, accurately estimating the optimal CoT length in real-world scenarios remains challenging due to the complexity and variability of reasoning tasks. While our theoretical and empirical analyses offer practical heuristics for coarse approximation, they may not fully capture the nuances of diverse problem domains or model behaviors. We leave the development of more precise estimation methods and adaptive strategies for optimal CoT length selection in complex, real-world settings as promising directions for future work.

B Formal Definitions of Simplified Arithmetic Problem

To begin, we aim to empirically investigate the relationship between reasoning performance and CoT length. Therefore, we need to control a given model to generate reasoning chains of varying lengths for a specific task. Unfortunately, no existing real-world dataset or model fully meets these strict requirements. Real-world reasoning tasks, such as GSM8K or MATH [9, 18], do not provide multiple solution paths of different lengths, and manually constructing such variations is challenging. Moreover, it is difficult to enforce a real-world model to generate a diverse range of reasoning paths for a given question. Given these limitations, we begin our study with experiments on synthetic datasets.

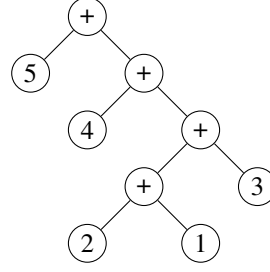


Figure 5: Computation tree of arithmetic expression $5 + (4 + ((2 + 1) + 3))$.

B.1 Problem Formulation

To investigate the effect of CoT length in a controlled manner, we design a synthetic dataset of simplified arithmetic tasks with varying numbers of reasoning steps in the CoT solutions.

Definition B.1 (Problem). In a simplified setting, an arithmetic task q is defined as a binary tree of depth T . The root and all non-leaf nodes are labeled with the $+$ operator, while each leaf node contains a numerical value (mod 10). In addition, we impose a constraint that every non-leaf node must have at least one numerical leaf as a child.

The bidirectional conversion method between arithmetic expressions and computation trees is as follows: *keeping the left-to-right order of numbers unchanged, the computation order of each "+" or tree node is represented by tree structure or bracket structures*. For example, consider the task $5 + (4 + ((2 + 1) + 3))$ with $T = 4$. The corresponding computation tree is defined as Figure 5.

To ensure that CoT solutions of the same length have equal difficulty for a specific problem, we assume that each reasoning step performs the same operations within a single CoT process.

Definition B.2 (Solution). We define a t -hop CoT with a fixed each step length of t as a process that executes t operations starting from the deepest level and moving upward recursively.

According to this definition, the execution sequence is uniquely determined. For example, one way to solve expression in Figure 5 is by performing one addition at a time:

$$5 + (4 + ((2 + 1) + 3)) = \langle 1 \rangle \quad (3)$$

$$2 + 1 = 3 \quad (4)$$

$$3 + 3 = 6$$

$$4 + 6 = 0$$

$$5 + 0 = 5 \langle \text{END} \rangle.$$

Another approach is to perform two additions at a time:

$$5 + (4 + ((2 + 1) + 3)) = \langle 2 \rangle \quad (5)$$

$$(2 + 1) + 3 = 6$$

$$5 + (4 + 6) = 5 \langle \text{END} \rangle.$$

The latter approach is half as long as the former, but each reasoning step is more complex⁶. This illustrates a clear trade-off between the difficulty of each subtask and the total number of reasoning steps.

⁶This is because performing two operations at once requires the model to either memorize all combinations of numbers in a two-operator equation and their answers, apply techniques like commutativity to reduce memory requirements, or use its mental reasoning abilities to perform the two operations without relying on CoT.

In practice, when t does not evenly divide T , the final step performs $T \bmod t$ operations. To guide the model in generating the desired CoT length, we insert the control token `<t>` after the question and before the beginning of the solution. To preserve the parentheses that indicate the order of operations, we construct expressions in Polish notation. However, for readability, we present each problem in its conventional form throughout the article.

B.2 Contrast to vanilla arithmetic problem

Why pruning? Initially, we intended to create a synthetic dataset for regular arithmetic tasks, but we quickly realized that the computation tree for such tasks is uncontrollable. For example, consider the task $1 * 2 + 3 * 4$. We hoped to compute 2 operators in one step, but found it impossible because the addition needs to be computed after the two multiplications, and we cannot aggregate two multiplications in one subtask. Therefore, pruning the computation tree becomes essential.

Why only focusing on addition? There are two reasons why we focus on arithmetic tasks involving only addition: first, it simplifies pruning, as the order of operations can be controlled solely by parentheses; second, it facilitates the computation of sub-tasks, since parentheses do not affect the final result, and the model only needs to compute the sum of all the numbers when solving a sub-task. We aim for the model to handle longer sub-tasks, thereby allowing a broader study of the impact of CoT length.

Will the simplified synthetic dataset impact the diversity of the data? We need to clarify that even with pruning, the structure of the expressions will still vary because swapping the left and right child nodes of each non-leaf node in the computation tree results in different expressions. When $T > 30$, the number of possible variations exceeds 1×10^9 .

C Supplementary Details on Real world Experiment for Optimal CoT Length

C.1 Implementation Details

Solution Length Control. To study the impact of CoT length on performance under a given problem difficulty, we need to induce the model to naturally generate solutions of varying lengths. Simply adding prompts like “*please use 100 tokens to solve this problem*” or “*please use 10 steps to solve this problem*” is not ideal because the model’s ability to follow instructions regarding output length is limited, and such fixed-length prompts may not ensure fairness across problems of different difficulties. Moreover, prompting for a specific length might lead the model to generate irrelevant tokens or steps just to “pad the length,” without actually changing the number of steps or the complexity of the reasoning. Additionally, controlling `max_length` is also problematic, as overly long responses might get truncated, which would directly lead to lower accuracy for longer outputs. What we really want is for the model to generate a complete and coherent long response on its own, so we can observe the corresponding accuracy.

To create solutions with varying step lengths with different complexity, we follow [12] by using in-context examples (8-shots) with three different levels of complexity to guide the model in generating solutions with different step counts. For each set of in-context examples, we sample 20 times, resulting in a total of 60 samples per question.

Step Segmentation. Simply measuring CoT length by counting tokens is neither rigorous nor meaningful. Since our focus is on final performance rather than efficiency, we care more about using CoT length to reflect the complexity of the reasoning pattern. In this sense, the number of reasoning steps can serve as a more appropriate indicator of CoT length. As we discussed earlier, the step number captures how the model decomposes the problem, which directly reflects the complexity of its reasoning. In contrast, token length fails to capture this because, as the model thinks more deeply and the number of steps increases, the number of tokens per step may decrease—making the total token count unpredictable and unreliable as a proxy for reasoning complexity.

When calculating the number of steps, we separate the full reasoning chain using “\n” [12] and remove empty lines caused by “\n\n”. Then we consider the total number of lines as the CoT length. Since questions in the MATH dataset are challenging and lead to high variability in final CoT lengths, we normalize the lengths by applying $\text{length} = \text{length} // \text{bin_width}$. For experiments comparing different models (e.g., optimal CoT length per model or optimal vs. longest CoT), the

questions within each length bin differ (though only 30 per group), which introduces variability. To reduce this variance and ensure each bin has enough samples, we use a relatively large bin width of 5. In contrast, for analyzing the influence of task difficulty, where each calculation on optimal CoT length only contains one question, we adopt a finer bin width of 2 for better resolution (we also verified that using width 1 yields almost identical results).

More Details of Figure 2b. When evaluating the results, questions with accuracy < 0.01 or > 0.99 (indicating all incorrect or all correct responses) are excluded, as their accuracy does not vary with step length changes.

To better understand the reliability of the observed trend between task difficulty and optimal Chain-of-Thought (CoT) length, we compute a 95% confidence interval around the linear regression line. Specifically, we use standard methods based on the Student’s t-distribution to estimate uncertainty in the predicted values. The confidence band reflects how much the estimated mean CoT length is expected to vary given the finite sample size and the distribution of data points.

C.2 More Experimental Results

Additional results for Figure 2b. To further investigate the relationship between task difficulty and optimal CoT lengths on real world datasets, we conduct experiments on different models. The results (Figure 6 and 7) are impressive that results on all models show a significant correlation between the task difficulties and optimal lengths.

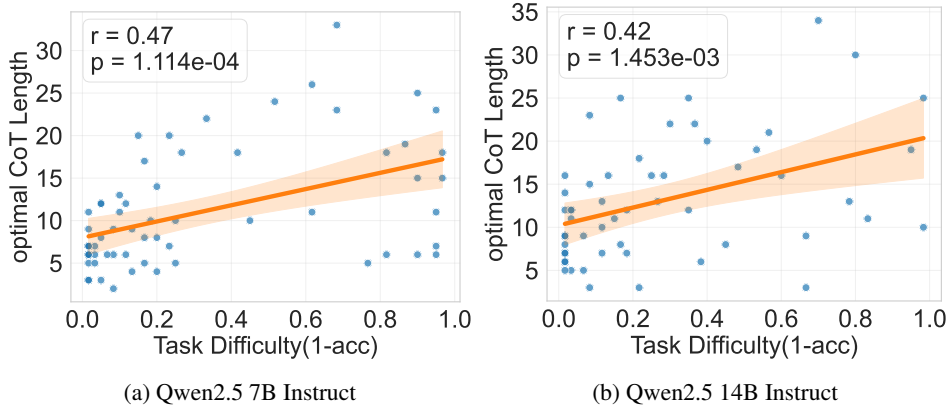


Figure 6: Evaluation between task difficulties and optimal CoT lengths on MATH datasets with Qwen2.5 Series Instruct models.

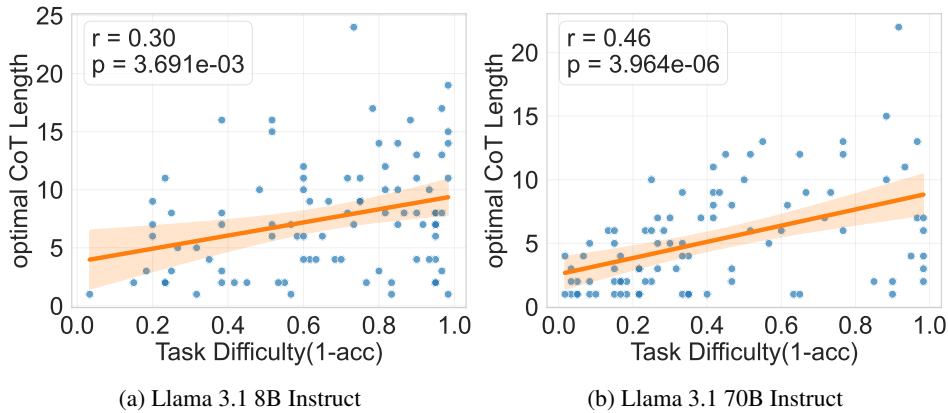
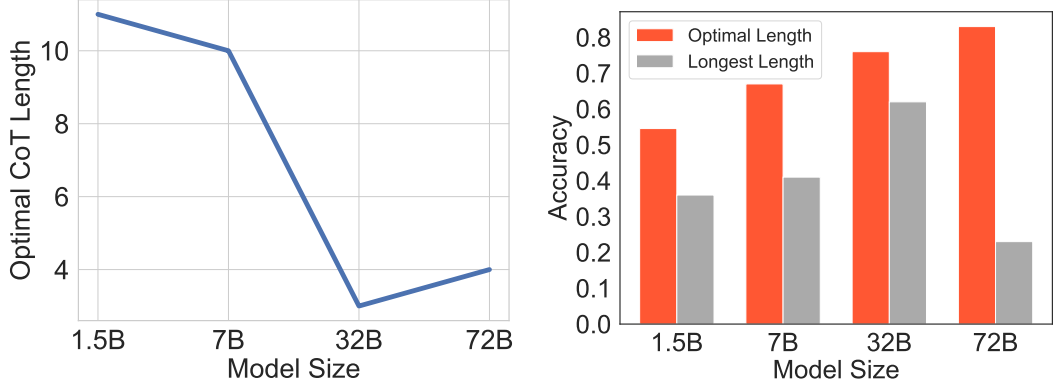


Figure 7: Evaluation between task difficulties and optimal CoT lengths on MATH datasets with LLama3.1 Series Instruct models.

Results on MMLU STEM dataset. We also conduct experiments on the MMLU STEM dataset using the Qwen2.5 Series instruct models under the same settings as the MATH dataset. The results, shown in Figures 8 and 9, exhibit similar trends to those observed on the MATH dataset.



(a) Optimal CoT length vs. Model size (Qwen2.5 series). (b) Optimal vs. Longest CoT length accuracy on MMLU STEM dataset.

Figure 8: Real-world CoT length observations. (a) Larger models tend to achieve optimal performance with shorter CoTs. (b) Accuracy for CoTs of optimal length is significantly higher than that of the longest CoTs.

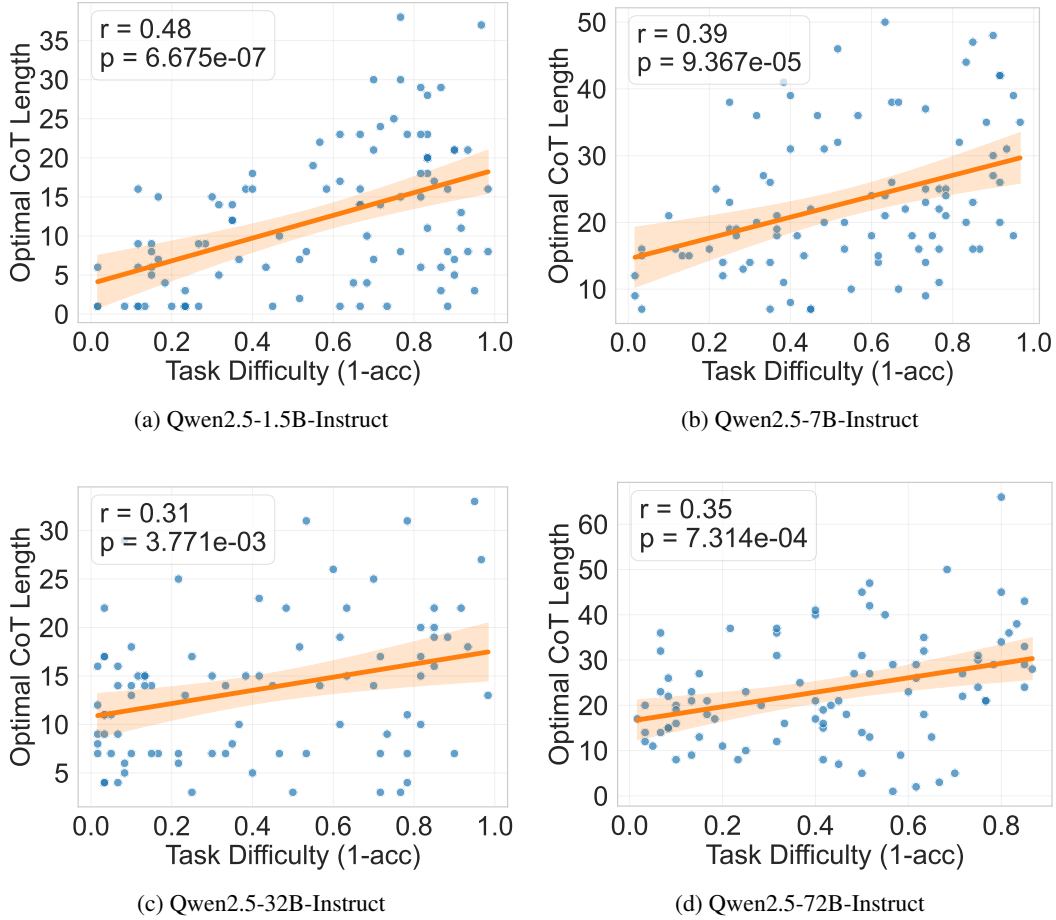


Figure 9: Evaluation between task difficulties and optimal CoT lengths on MMLU STEM datasets with Qwen2.5 Series Instruct models.

D Supplementary Details on Real World Experiment for RL Simplicity Bias

For Figure 1b, we use Qwen2.5-7B-Instruct [27] as the base model, Group Relative Policy Optimization with R1-like prompting [31, 15] for the reinforcement learning process, and LeetCode-2K [40] as the training and evaluation dataset. We take the following training configuration by default:

Table 1: Hyperparameter settings for real-world RL experiments with Qwen2.5-instruct models.

| Learning Rate | Max Epochs | Rollout Samples | Reverse KL Coefficient | Entropy Loss Coefficient | Effective Batch Size |
|---------------|------------|-----------------|------------------------|--------------------------|----------------------|
| 5e-7 | 10 | 16 | 1e-3 | 5e-3 | 256 |

E Additional Synthetic Experiment Details

E.1 Training details

In default, we train different models (layers ranging from 5 to 9) on the same dataset, which included mixed questions with total operators $T \in [12, 80]$ and random sampled CoT solutions with each step operators $t \in [1, 12]$. All other parameters are kept the same with the huggingface GPT-2 model. During the training process, the CoT indicator token $\langle \tau \rangle$ is also trained, so that during test-time, we can let the model decide which type of CoT it will use by only prompting the model with the question. For each model, we train 25000 iterations with batch size that equals 256. During test-time, we test 100 questions for each T and t . All experiments can be conducted on one NVIDIA A800 80G GPU.

E.2 Observation of subtask loss

As we observed in training losses, the loss of subtask generation tokens (e.g. $1 + 2$) for the easiest subtask ($t = 1$) is about 3 times larger than the hardest subtask ($t = 12$), while the loss ratio for subtask answer tokens is $1e4$. Therefore, it is acceptable for taking the subtask error rate constant with t .

Besides, there is no obvious pattern showing the model sizes affect the subtask loss. Moreover, the smallest model and the largest model have almost the same subtask loss. Therefore, in our settings, we take model size as irrelevant with the subtask error rate.

F Theoretical Results under Broader Scenarios

F.1 General Errors

In the simple case we discussed in Section 4.1, we discussed the trend of overall accuracy with respect to N and the variation of optimal N with M and T , assuming the subtask error rate is a linear function. In the following discussion, we aim to derive conclusions corresponding to more general error rate functions. We find that as long as the error function satisfies some basic assumptions on the **monotonicity** and **convexity** of the error functions, the above conclusions still hold.

Assumption F.1. $E(N, M, T)$ satisfies the following reasonable conditions:

- $0 < E(N = 1, M, T) < 1$
- $\lim_{N \rightarrow +\infty} E(N, M, T) = 0$
- $E(N, M, T)$ is monotonically decreasing with N , since more detailed decomposition leads to easier subtask.
- $E(N, M, T)$ is convex with N , since the benefits of further decomposing an already fine-grained problem (N is large) are less than the benefits of decomposing a problem that has not yet been fully broken down (N is small).
- $E(N, M, T)$ is monotonically decreasing with M , since stronger models have less subtask error rate.
- $E(N, M, T)$ is monotonically increasing with T , since harder total task leads to harder subtask while N, M are the same.

Assumption F.2. $\sigma(T)$ is monotonically increasing with T

With Assumption F.1 and F.2), the core insights from the linear case can be generalized.

Theorem F.3. *For a noise function $0 < \sigma(T) < 1$ and a subtask error rate function $0 < E(N, M, T) < 1$ satisfying Assumptions F.1 and F.2, the general final accuracy function $A(N)$ from Proposition 4.2 has the following properties:*

- $\lim_{N \rightarrow +\infty} A(N) = 0$. (Excessively long chains always fail.)
- If $A(N)$ has a maximum at $N^* > 1$, then N^* has a lower bound related to M and T :

$$N^* \geq N_{LB}(M, T) = E_N^{-1} \left(1 - \frac{1}{e^2(1 - \sigma(T))}; M, T \right), \quad (6)$$

where $E_N^{-1}(\cdot; M, T)$ is the inverse of $E(N, M, T)$ with respect to N .

The monotonicity of E_N^{-1} with respect to M (decreasing) and T (increasing, assuming $\sigma(T)$ doesn't dominate adversely) implies that the qualitative scaling laws (Corollaries stemming from Theorem 4.3) still hold under general conditions, supporting the empirically observed Simplicity Bias and the inverted U-shaped performance.

Corollary F.4. *As the model becomes stronger, E^{-1} decreases monotonically with respect to M , which leads to a decrease of $N(M, T)$.*

Corollary F.5. *As the task becomes harder, E^{-1} is monotonically increasing with respect to T , which leads to an increase in $N(M, T)$.*

F.2 Random Error

In Theorem 4.3 and F.3, we make a strong assumption that all sub-question or sub-answer errors are identical, which does not align well with real-world scenarios. In practice, each sub-task may exhibit a different error rate. However, they generally follow a trade-off: the more the task is decomposed, the easier each sub-task becomes. Specifically, we can model the error rate of each sub-task as a random variable with a fixed expectation that monotonically decreases with the number of CoT steps N .

To simplify the problem, here we assume $\sigma_i \sim B(\alpha_1(T), \beta_1(T))$ to be the sub-question error rate, and $e_i \sim B(\alpha_2(N, M, T), \beta_2(N, M, T))$ to be the sub-answer error rate. Then, as a variant of Proposition 4.2, the expectation of final accuracy is $\mathbb{E} \left[\prod_{i=1}^N (1 - e_i)(1 - \sigma_i) \right]$.

It is worth noting that each σ_i or e_i is not independent. If most steps are easy (i.e., have low error rates), the remaining steps are more likely to be easy as well. Moreover, if a particular step serves as a self-validation step, its high accuracy can influence the correctness of other steps that depend on it. This also provides an interpretation for reasoning models exhibiting backtracking behavior.

Theorem F.6. *Let $\alpha_1 = T$, $\beta_1 = C - T$, $\alpha_2 = T$, and $\beta_2 = NM - T$. Then the expected error rates for sub-questions and sub-answers are given by $\mathbb{E}[\sigma_i] = \frac{T}{C}$ and $\mathbb{E}[e_i] = \frac{T}{MN}$, respectively. Based on these estimates, we can derive an upper bound $\hat{A}(N)$ on the final accuracy*

$$\mathbb{E} \left[\prod_{i=1}^N (1 - e_i)(1 - \sigma_i) \right] \leq \hat{A}(N) = \left[\left(1 - \frac{T}{C + 2N - 1} \right) \left(1 - \frac{T}{NM + 2N - 1} \right) \right]^N,$$

which initially increases and then decreases as the number of CoT steps N grows.

This suggests that even with stochasticity, the fundamental trade-off leading to an optimal CoT length persists.

G Proof

In this section, we provide the proofs for all theorems.

G.1 Proof of Proposition 4.2

Proposition 4.2. *The total accuracy of N -step reasoning is*

$$A(N) = P(a_{\text{final}} = a_{\text{final}}^* | q, \theta, N) = \alpha ((1 - E(N, M, T))(1 - \sigma(T)))^N, \quad (1)$$

where α denotes a constant value independent of N .

Proof. In each subtask t_i , which contains t operators, there are $2t + 1$ tokens (as the number of numerical tokens is one more than the number of operators). Therefore, the accuracy of each subtask is given by

$$P(t_i = t_i^* | H_{i-1}, q, \theta) = (1 - \sigma(T))^{2t+1}. \quad (7)$$

In our theoretical analysis, for simplicity, we allow t to be a fraction, defined as $t = \frac{T}{N}$, and assume that each subtask has the same level of difficulty given T and N . Under this assumption, we have the final accuracy:

$$A(N) = P(a_N = a_N^* | q, \theta) \quad (8)$$

$$= \prod_{i=1}^N P(t_i = t_i^* | H_{i-1}, q, \theta) P(a_i = a_i^* | t_i, H_{i-1}, q, \theta) \quad (9)$$

$$= \prod_{i=1}^N (1 - \sigma(T))^{2t+1} (1 - E(N, M, T)) \quad (10)$$

$$= (1 - \sigma(T))^{N(2t+1)} (1 - E(N, M, T))^N \quad (11)$$

$$= (1 - \sigma(T))^{2T} ((1 - E(N, M, T))(1 - \sigma(T)))^N \quad (12)$$

$$= \alpha ((1 - E(N, M, T))(1 - \sigma(T)))^N \quad (13)$$

□

G.2 Proof of Theorem 4.3

Theorem 4.3 (Optimal CoT Length). *For a given model capability M and task difficulty T , the total accuracy $A(N) = \alpha[(1 - T/C) \cdot (1 - T/(NM))]^N$ (Eq. (1)) initially increases and then decreases as N increases (forming an inverted U-shape). Thus, there exists an optimal CoT length:*

$$N^*(M, T) = \frac{TZ}{M(Z + 1)}, \quad (2)$$

that maximizes $A(N)$, where $Z = W_{-1}(-1 - T/(Ce))$, and $W_{-1}(x)$ is the smaller real branch of the Lambert W function satisfying $we^w = x$, and e is the natural number.

Proof. Given Eq. (1) that

$$A(N) = \alpha \left(\left(1 - \frac{T}{C}\right) \left(1 - \frac{T}{NM}\right) \right)^N \quad (14)$$

We consider function

$$f(x) = \left[\left(1 - \frac{T}{Mx}\right) \left(1 - \frac{T}{C}\right) \right]^x. \quad (15)$$

For convenience, define

$$g(x) = \ln(f(x)) = x \ln \left[\left(1 - \frac{T}{Mx}\right) \left(1 - \frac{T}{C}\right) \right].$$

Thus,

$$g'(x) = \left[\ln \left(1 - \frac{T}{Mx}\right) + \frac{T}{Mx \left(1 - \frac{T}{Mx}\right)} \right] + \ln \left(1 - \frac{T}{C}\right).$$

Set $g'(x) = 0$:

$$\ln\left[\left(1 - \frac{T}{Mx}\right)\left(1 - \frac{T}{C}\right)\right] + \frac{T}{Mx\left(1 - \frac{T}{Mx}\right)} = 0.$$

Let $A = \frac{1}{1 - \frac{T}{Mx}}$, then we have

$$\ln\left[\left(1 - \frac{T}{C}\right)\right] + A - 1 = \ln(A).$$

Let $z := 1 - T/C$. (Since $T/C < 1$, $z = 1 - T/C > 0$.) By moving terms, we have:

$$-\frac{z}{e} = -A \exp(-A).$$

Therefore,

$$A = -W^{-1}\left(-\frac{z}{e}\right) = -Z,$$

Finally, we have

$$N(M, T) = x = \frac{TZ}{M(Z + 1)}$$

Here $W(\cdot)$ is the **Lambert W function**, and for $0 < 1 - \frac{T}{C} < 1$, the argument $\alpha = -\frac{1-T/C}{e}$ lies in the interval $(-\frac{1}{e}, 0)$. This means there are two real branches W_0 and W_{-1} in that domain, but since $\frac{Z}{Z+1} > 0$, we have $Z < -1$. Therefore, we only take the solution on branch W_{-1} . \square

G.3 Proof of Corollary 4.4

Corollary 4.4 (Scaling laws of Optimal CoT Length). *Based on Theorem 4.3, one can derive:*

- $N^*(M, T)$ increases monotonically with T , i.e., harder tasks require more reasoning steps to attain the optimal performance.
- The optimal number of operators per step $t^* = T/N^*(M, T) = M(1 + 1/Z)$ increases monotonically with T . This aligns with the envelope curve result (Figure 3a).
- $N^*(M, T)$ decreases monotonically with M , i.e., more capable models require fewer reasoning steps to attain the optimal performance, reflecting the simplicity bias.

Proof. The second and third conclusions can be easily derived through monotonic composition, so we primarily focus on proving the first point. We begin the proof by incorporating the notation from G.2. We have

$$g'(x) = \left[\ln\left(1 - \frac{T}{Mx}\right) + \frac{T}{Mx\left(1 - \frac{T}{Mx}\right)} \right] + \ln\left(1 - \frac{T}{C}\right),$$

and $x^*(T)$ such that $g'(x^*(T)) = 0$.

Let $F(x^*(T), T) = g'(x^*(T)) = 0$. We want to see how $x^*(T)$ changes as T changes, therefore we take total derivative w.r.t. T . By the chain rule,

$$0 = \frac{d}{dT} F(x^*(T), T) = \underbrace{\frac{\partial F}{\partial x}(x^*(T), T)}_{\text{call this } F_x} \cdot \frac{\partial x^*}{\partial T}(T) + \underbrace{\frac{\partial F}{\partial T}(x^*(T), T)}_{\text{call this } F_T}.$$

Hence

$$\frac{\partial x^*}{\partial T}(T) = - \frac{F_T(x^*(T), T)}{F_x(x^*(T), T)}.$$

So the sign of $x'^*(T)$ is the opposite of the sign of F_T , provided $F_x \neq 0$.

Since

$$F_x(x, T) = -\frac{T^2}{x(Mx - T)^2} < 0, \forall x > 0, \quad (16)$$

all we need to prove is

$$F_T(x^*(T), T) = \frac{T}{(Mx^*(T) - T)^2} - \frac{1}{C - T} > 0. \quad (17)$$

That is

$$\frac{\sqrt{T(C - T)} + T}{M} > x^*(T). \quad (18)$$

Let $x_0(T) = \frac{\sqrt{T(C - T)} + T}{M}$ be the test point.

According to Lemma G.1, $F(x_0(T), T) < 0$. Since $F(x^*(T), T) = 0$, and $F_x(x^*(T), T) < 0$, we have $x_0(T) > x^*(T)$.

Thus, $F_T(x^*(T), T) > 0$ holds and we have proved our corollary with $\frac{\partial x^*}{\partial T}(T) > 0$. □

G.4 Proof of Theorem F.3

Theorem F.3. *For a noise function $0 < \sigma(T) < 1$ and a subtask error rate function $0 < E(N, M, T) < 1$ satisfying Assumptions F.1 and F.2, the general final accuracy function $A(N)$ from Proposition 4.2 has the following properties:*

- $\lim_{N \rightarrow +\infty} A(N) = 0$. (Excessively long chains always fail.)
- If $A(N)$ has a maximum at $N^* > 1$, then N^* has a lower bound related to M and T :

$$N^* \geq N_{LB}(M, T) = E_N^{-1}\left(1 - \frac{1}{e^2(1 - \sigma(T))}; M, T\right), \quad (6)$$

where $E_N^{-1}(\cdot; M, T)$ is the inverse of $E(N, M, T)$ with respect to N .

Proof. (1) Since $0 < A(N) < (1 - \sigma(T))^N$, and $\lim_{N \rightarrow +\infty} (1 - \sigma(T))^N = 0$, $\lim_{N \rightarrow +\infty} A(N, M, T) = 0$

(2) Let $g(x)$ denote $E(x, M, T)$ and define $f(x) = \ln A(x)$. Then,

$$f'(x) = \ln(1 - \sigma(T)(1 - g(x))) - \frac{x E'(x)}{1 - E(x)} \quad (19)$$

$$< \ln(1 - \sigma(T)(1 - g(x))) + 2, \quad (\text{since } E \text{ is convex and } x = N \geq 1) \quad (20)$$

If $A(N)$ attains its maximum at some point $N^* > 1$, then $\ln(1 - \sigma(T)) + 2 > 0$. Otherwise, we would have $f'(x) < \ln(1 - \sigma(T)) + 2 \leq 0 \forall x > 1$, leading to a contradiction.

Thus, it follows that $e^2(1 - \sigma(T)) > 1$.

Now, define $N(M, T) = E^{-1}\left(1 - \frac{1}{e^2(1 - \sigma(T))}\right)$, which satisfies

$$\ln(1 - \sigma(T)(1 - g(N(M, T)))) + 2 = 0.$$

If there exists $x^* < N(M, T)$ such that $f'(x^*) = 0$, then we obtain

$$0 = f'(x^*) < \ln(1 - \sigma(T)(1 - E(x^*))) + 2 < 0,$$

which is a contradiction. Hence, the assumption that $x^* < N(M, T)$ must be false.

Therefore, we conclude that $x^* = N^* > N(M, T)$. □

G.5 Proof of Theorem F.6

Theorem F.6. Let $\alpha_1 = T$, $\beta_1 = C - T$, $\alpha_2 = T$, and $\beta_2 = NM - T$. Then the expected error rates for sub-questions and sub-answers are given by $\mathbb{E}[\sigma_i] = \frac{T}{C}$ and $\mathbb{E}[e_i] = \frac{T}{MN}$, respectively. Based on these estimates, we can derive an upper bound $\hat{A}(N)$ on the final accuracy

$$\mathbb{E} \left[\prod_{i=1}^N (1 - e_i)(1 - \sigma_i) \right] \leq \hat{A}(N) = \left[\left(1 - \frac{T}{C + 2N - 1}\right) \left(1 - \frac{T}{NM + 2N - 1}\right) \right]^N,$$

which initially increases and then decreases as the number of CoT steps N grows.

Proof. According to the multidimensional version of Hölder's inequality,

$$\mathbb{E} \left[\prod_{i=1}^N (1 - e_i)(1 - \sigma_i) \right] \leq \prod_{i=1}^N (\mathbb{E}[(1 - e_i)^{2N}] \mathbb{E}[(1 - \sigma_i)^{2N}])^{\frac{1}{2N}} \quad (21)$$

$$\stackrel{\text{(Lemma G.2)}}{\leq} \prod_{i=1}^N \left(1 - \frac{T}{C + 2N - 1}\right) \left(1 - \frac{T}{NM + 2N - 1}\right) \quad (22)$$

$$= \left[\left(1 - \frac{T}{C + 2N - 1}\right) \left(1 - \frac{T}{NM + 2N - 1}\right) \right]^N \quad (23)$$

□

G.6 Proof of Corollary 4.5

Corollary 4.5 (RL Converges to Optimal CoT Length). *For gradient ascent on $J(\theta)$ with sufficiently small step size, the policy converges to a deterministic policy $\pi_\theta(N_i) = 1$ iff $i = \arg \max_j A(N_j)$. Thus, RL training converges to the optimal CoT length $N^* = \arg \max_{N \in \mathcal{A}} A(N)$.*

Proof. We treat the choice of CoT length as a k -armed stochastic bandit with action set $\mathcal{A} = \{N_1, \dots, N_k\}$ and unknown success probabilities⁷ $A(N_i) \in (0, 1)$. Without loss of generality, relabel the arms so that

$$A(N_1) = \max_j A(N_j) =: A^*, \quad A(N_1) \geq A(N_2) \geq \dots \geq A(N_k).$$

The agent uses a softmax (Gibbs) policy

$$\pi_\theta(N_i) = \frac{e^{\theta_i}}{\sum_{j=1}^k e^{\theta_j}}, \quad \theta \in \mathbb{R}^k, \quad (24)$$

and maximises the expected reward

$$J(\theta) = \sum_{i=1}^k \pi_\theta(N_i) A(N_i). \quad (25)$$

Because π_θ is C^∞ in θ and $A(N_i)$ are constants, J is smooth.

Under the REINFORCE estimator with sufficiently small, fixed step size $\eta > 0$, gradient ascent updates take the form

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta J(\theta^{(t)}), \quad (26)$$

where

$$\frac{\partial J}{\partial \theta_i} = \pi_\theta(N_i) (A(N_i) - J(\theta)). \quad (27)$$

⁷By Proposition 4.2, $A(N_i)$ is the probability that the final answer is correct when a chain of length N_i is used. The bandit is *stationary* because $A(N_i)$ does not depend on time or the agent's past actions.

Eq. (27) is the classical *replicator* (or logit) gradient. Define the simplex $\Delta^{k-1} := \{\pi \in (0, 1]^k \mid \sum_i \pi_i = 1\}$ and write $\pi_\theta = (\pi_\theta(N_1), \dots, \pi_\theta(N_k))$.

Letting $\eta \rightarrow 0$ yields the ODE

$$\dot{\pi}_i = \pi_i (A(N_i) - \langle \pi, A \rangle), \quad i = 1, \dots, k, \quad (28)$$

with $\langle \pi, A \rangle = \sum_j \pi_j A(N_j)$. Eq. (28) is the **replicator dynamics** for a fitness landscape A on Δ^{k-1} .

Consider the Kullback–Leibler divergence to the optimal pure strategy $\mathbf{e}_1 = (1, 0, \dots, 0)$,

$$V(\pi) = \sum_{i=1}^k \pi_i \ln\left(\frac{\pi_i}{e_{1,i}}\right) = -\ln \pi_1.$$

V is non-negative on Δ^{k-1} and $V(\pi) = 0$ iff $\pi = \mathbf{e}_1$.

Taking the time derivative along Eq. (28) gives

$$\frac{dV}{dt} = -\frac{\dot{\pi}_1}{\pi_1} = -(A(N_1) - \langle \pi, A \rangle) \leq 0,$$

with equality iff $\pi_1 = 1$ or $A(N_1) = \langle \pi, A \rangle$. The latter can only happen if $\pi_1 = 1$ because $A(N_1) > A(N_j)$ for $j > 1$. Hence V is a strict Lyapunov function, and \mathbf{e}_1 is the *unique* asymptotically stable equilibrium of Eq. (28). All other stationary points (mixtures over sub-optimal arms) are unstable.

For sufficiently small but fixed η (choose $\eta < \frac{1}{A^*}$, which always exists), projected gradient ascent is a *perturbed* discretisation of Eq. (28). Standard results for primal-space mirror descent imply that the discrete iterates $\pi^{(t)} \equiv \pi_{\theta(t)}$ converge almost surely to the set of asymptotically stable equilibria of the ODE, i.e. to $\{\mathbf{e}_1\}$. Therefore

$$\lim_{t \rightarrow \infty} \pi_{\theta(t)}(N_i) = \begin{cases} 1, & \text{if } i = \arg \max_j A(N_j), \\ 0, & \text{otherwise.} \end{cases}$$

Because A may attain its maximum at several arms, the limit is a deterministic policy that places all probability on *some* maximiser of A .

Thus gradient ascent on Eq. (25) converges to a deterministic policy that always selects an optimal CoT length $N^* = \arg \max_{N \in \mathcal{A}} A(N)$, completing the proof. \square

G.7 Technical Lemmas

Lemma G.1 (test point). *Let $F(x)$ be defined as*

$$F(x) = \ln\left(1 - \frac{T}{Mx}\right) + \frac{T}{Mx\left(1 - \frac{T}{Mx}\right)} + \ln\left(1 - \frac{T}{C}\right),$$

where $T, M, C \in \mathbb{R}^+$ satisfy the conditions:

- $0 < \frac{T}{C} < 0.9$,
- $0 < \frac{T}{Mx} < 1$.

Define x_0 as

$$x_0 = \frac{\sqrt{T(C-T)} + T}{M}.$$

Then, we have

$$F(x_0) < 0.$$

Proof. At $x = x_0$, note that

$$Mx_0 = \sqrt{T(C-T)} + T.$$

Thus,

$$1 - \frac{T}{Mx_0} = 1 - \frac{T}{T + \sqrt{T(C-T)}} = \frac{\sqrt{T(C-T)}}{T + \sqrt{T(C-T)}}.$$

Therefore,

$$\ln\left(1 - \frac{T}{Mx_0}\right) = \ln\left(\frac{\sqrt{T(C-T)}}{T + \sqrt{T(C-T)}}\right) = \ln\sqrt{T(C-T)} - \ln(T + \sqrt{T(C-T)}).$$

Also, observe that

$$\frac{T}{Mx_0\left(1 - \frac{T}{Mx_0}\right)} = \frac{T}{(T + \sqrt{T(C-T)})\left(\frac{\sqrt{T(C-T)}}{T + \sqrt{T(C-T)}}\right)} = \frac{T}{\sqrt{T(C-T)}} = \sqrt{\frac{T}{C-T}}.$$

It is convenient to introduce the change of variable

$$u = \sqrt{\frac{T}{C-T}},$$

so that

$$T = u^2(C-T), \quad \sqrt{T(C-T)} = u(C-T).$$

Then we have

$$T + \sqrt{T(C-T)} = u^2(C-T) + u(C-T) = u(C-T)(u+1).$$

In these terms we have:

$$\ln\sqrt{T(C-T)} = \ln[u(C-T)] = \ln u + \ln(C-T),$$

$$\ln(T + \sqrt{T(C-T)}) = \ln[u(C-T)(u+1)] = \ln u + \ln(C-T) + \ln(u+1),$$

and

$$\sqrt{\frac{T}{C-T}} = u.$$

Finally, we have

$$\ln\left(1 - \frac{T}{C}\right) = -\ln\left(\frac{C}{C-T}\right) = -\ln(u^2 + 1)$$

Thus, the function $F(x_0)$ becomes

$$F(x_0) = \ln u + \ln(C-T) - (\ln u + \ln(C-T) + \ln(u+1)) + u - \ln(u^2 + 1) \quad (29)$$

$$= -\ln(u+1) + u - \ln(u^2 + 1), \quad (30)$$

where $u = \sqrt{\frac{T}{C-T}} \in (0, 3)$. It is easy to show $F(x_0) < 0$ when $u \in (0, 3)$. □

Lemma G.2 (Estimation of the n -th Moment of the Beta Distribution). *Let $x \sim \text{Beta}(\alpha, \beta)$. Then*

$$\mathbb{E}[(1-x)^n] \leq \left(1 - \frac{\alpha}{\alpha + \beta + n - 1}\right)^n.$$

Proof.

$$\begin{aligned}
\mathbb{E}[(1-x)^n] &= \frac{1}{B(\alpha, \beta)} \int_0^1 (1-x)^n x^{\alpha-1} (1-x)^{\beta-1} dx \\
&= \frac{1}{B(\alpha, \beta)} \int_0^1 x^{\alpha-1} (1-x)^{\beta+n-1} dx \\
&= \frac{B(\alpha, \beta+n)}{B(\alpha, \beta)} \\
&= \frac{\Gamma(\alpha)\Gamma(\beta+n)}{\Gamma(\alpha+\beta+n)} \cdot \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \\
&= \frac{\Gamma(\beta+n)}{\Gamma(\beta)} \cdot \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha+\beta+n)} \\
&= \prod_{i=0}^{n-1} \frac{\beta+i}{\alpha+\beta+i} \\
&\leq \left(\frac{\beta+n-1}{\alpha+\beta+n-1} \right)^n \\
&= \left(1 - \frac{\alpha}{\alpha+\beta+n-1} \right)^n.
\end{aligned}$$

□

H Pseudo-code of Length-filtered Vote

Algorithm 1 Length-filtered Vote

```

1: Input: Model  $f_\theta$ , Question  $q$ , Space of All Possible Answers  $A$ , Number of Total Groups  $M$ ,
   Number of Selected Groups  $K$ , Group Width  $D$ 
2: Output: Final Answer  $\hat{a}$ 
3: Sample candidates  $c_1, \dots, c_n \stackrel{i.i.d.}{\sim} f_\theta(q)$ 
4: Define  $\mathcal{A}(c)$  as the corresponding answer of candidates  $c$ .
5: Define  $p_j \in [0, 1]^{|A|}$  as the frequency of each answer in length group  $L_j$ .
6: for  $j = 1$  to  $m$  do
    $L_j = \{c_i \mid \ell(c_i) \in [D * (j-1), D * j], i = 1, \dots, n\}$ 
7:   for  $a \in A$  do
     
$$p_j[a] = \frac{\sum_{c \in L_j} \mathbb{I}(\mathcal{A}(c) = a)}{|L_j|}$$

8:   end for
9: end for
10:  $\{s_1, \dots, s_K\} = \arg \min_{S \subseteq \{1, \dots, M\}, |S|=K} \sum_{s \in S} H(p_s)$ 
11:  $\hat{a} = \arg \max_{a \in A} \sum_{c \in L_{s_1} \cup \dots \cup L_{s_K}} \mathbb{I}(\mathcal{A}(c) = a)$ 
12: return  $\hat{a}$ 

```
