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.

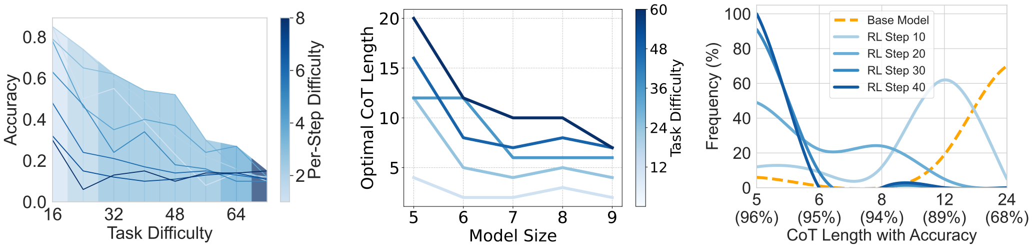
```
Problem: 1+2+3+4+5+6+7
Step 1: 1+2 = 3. (Remaining: 3+3+4+5+6+7)
Step 2: 3+3 = 6. (Remaining: 6+4+5+6+7)
...
Step 6: 21+7 = 28. (Final Answer)
```

- 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



(a) Optimal per-step difficulty (b) Optimal CoT vs. model size (c) Evolution of CoT during RL

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 $H_k := [t_1, a_1, \cdots, t_k, a_k]$ 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 $P(a_i = a^*|q_i,H_{i-1},q,\theta)$.

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, $P(a_i = a^*|q_i,H_{i-1},q,\theta)$ 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*

$$A(N) = P(a_{\textit{final}} = a^*_{\textit{final}}|q,\theta,N) = \alpha\left((1-E(N,M,T))(1-\sigma(T))\right)^N, \tag{1}$$

where α denotes a constant value independent of N.

[5] Note 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.